Solver Reference Manual

LUSAS Version 14.5 : Issue 1

LUSAS
Forge House, 66 High Street, Kingston upon Thames,
Surrey, KT1 1HN, United Kingdom

Tel: +44 (0)20 8541 1999
Fax +44 (0)20 8549 9399
Email: info@lusas.com
http://www.lusas.com

Distributors Worldwide

Copyright ©1982-2010 LUSAS
All Rights Reserved.

Table of Contents

Notation iii
Chapter 1 Introduction 1
Introduction 1
LUSAS Data Chapters 1
Chapter 2 Syntax and Data Preparation 7
Reading and Writing LUSAS Data Syntax 7
Chapter 3 LUSAS Data Input 13
System Parameters 13
Problem Definition 15
Data Groups 17
Element Topology 17
Solution Order 20
Solver Type 22
Nodal Coordinates 25
Geometric Properties 43
Composite Geometry 46
Geometric Assignments 47
Material Properties 47
Linear Material Properties 48
Matrix Properties 57
Joint Material Properties 58
Material Properties Mass 72
Nonlinear Material Properties 73
Field Material Properties 133
Material Assignments 139
Composite Material 140
Composite Assignments 142
Element Ages 142
Activate/Deactivate Elements 143
Damping Properties 152
Slidelines 153
Thermal Surfaces 169
Nodal Freedoms 188
Freedom Template 189
Cartesian Sets 190
Transformed Freedoms 193
Constraint Equations 194
Support Conditions 196
Coupled Analysis 199
Structural Loading 201
General Point/Patch Loads 223
Field Loading 228

Curve Definition 233
Retained Freedoms 239
Output Control 240
Load Combinations 246
Enveloping Results 247
Superelements 247
Analysis Control 268
Nonlinear Control 269
Dynamic Control 284
Transient Control 290
Viscous Control 293
Eigenvalue Control 301
Guyan Control 306
Modal Damping Control 308
Spectral Control 311
Harmonic Response Control 313
Fourier Control 316
Creating a Plot File 318
Restart Facilities 318
Re-Solution 321
End 322
Appendix A LUSAS User Options 323
LUSAS User Options 323
LUSAS User Options by Category 329
Appendix B Nonlinear Hardening Material Convention 335
Nonlinear Hardening Material Convention 335
Appendix C Material Model Interface 339
Solver User Interface Routines 339
Programming Rules 341
Declaration 342
User Defined Constitutive Models 343
User Defined Resultant Models 354
User Defined Joint Models 363
User Defined Creep Models 371
User Defined Damage Models 375
User Defined Friction Models 378
User-Defined Rate of Internal Heat Generation 381
Utility Routines 386

Notation

A	Cross sectional area
Ap	Plastic area
As, Asy, Asz	Effective shear area
A1 ... An	Nodal cross sectional areas
$\mathbf{a_r}$	Mass Rayleigh damping constant
α	Coefficient of thermal expansion
α_s	Softening parameter
$\alpha_x, \alpha_y, \alpha_z, \alpha_{xy}, \alpha_{xz}, \alpha_{yz}$	Orthotropic thermal expansion coefficients
$\alpha_x, \alpha_y, \alpha_z$	Angular accelerations (radians/sec^2)
$\mathbf{b_r}$	Stiffness Rayleigh damping parameter
β_r	Shear retention factor/parameter
β	Principal stresses direction (radians)
C	Specific heat capacity
$\mathbf{C_i}$	(i)th hardening stiffness
C0	Neo-Hookean rubber model constant
C1, C2	Mooney-Rivlin rubber model constants
c	Cohesion
co	Initial cohesion
$\mathbf{D_{ij}}$	Rigidity coefficients
du, dq	Relative displacement, rotation
E	Modulus of elasticity (Young's modulus)
Ep	Elasto-plastic modulus
$\mathbf{E_x, E_y, E_z}$	Orthotropic modulus of elasticity
$\mathbf{e_p}$	Strain at peak compressive strength
$\mathbf{e_y, e_z}$	Eccentricity
$\varepsilon_x, \varepsilon_y, \varepsilon_z$	Direct strains (local or global)
ε_s	Maximum shear strain
ε_e	Von Mises equivalent strain
ε_c	Creep strains
$\bar{p}$	Equivalent plastic strain
$\mathbf{F_x, F_y, F_z}$	Forces (local or global)
$\mathbf{F_{yld}}$	Yield force
F	Deformation gradient
$\mathbf{f_c'}$	Compressive strength of concrete
$\mathbf{f_t'}$	Tensile strength of concrete
ψ_x, ψ_y, ψ_z	Flexural strain resultants
$\psi_{xy}, \psi_{xz}, \psi_{yz}$	Torsional strain resultants
G	Shear modulus
Gf	Fracture energy
$\mathbf{G_{xy}, G_{xz}, G_{yz}}$	Orthotropic shear modulus values
$\gamma_x, \gamma_y, \gamma_z$	Membrane strain resultants

$\gamma_x, \gamma_y, \gamma_z$	Field gradients (local or global)
$\mathbf{H}$	Enthalpy
$\mathbf{H_{I1}}$	Isotropic hardening parameter
$\mathbf{H_{K1}}$	Kinematic hardening parameter
$\mathbf{h_c}$	Convective heat transfer coefficient
$\mathbf{h_r}$	Radiative heat transfer coefficient
$\theta_x, \theta_y, \theta_z$	Rotations (local or global)
θ_1, θ_2	Loof node rotations (local)
θ_a, θ_b	Nodal rotations for thick shells
$\theta\lambda$	Angle defining principal directions of l1, l2
$\mathbf{I_y}, \mathbf{I_z}$	1st moments of inertia
$\mathbf{I_{yy}}, \mathbf{I_{zz}}$	2nd moments of inertia
$\mathbf{I_{yz}}$	Product moment of inertia
$\mathbf{J}$	Volume ratio (determinant of F)
$\mathbf{K}$	Spring stiffness
$\mathbf{K_c}$	Contact stiffness
$\mathbf{K_l}$	Lift-off stiffness
$\mathbf{K_o}$	Original gap conductance
$\mathbf{K_t}$	Torsional constant
$\mathbf{k}$	Thermal conductivity
$\mathbf{k_x}, \mathbf{k_y}, \mathbf{k_z}$	Orthotropic thermal conductivity values
$\mathbf{k_r}$	Bulk modulus
κ	Hardening stiffness
$\mathbf{L_i}$	Limit of (i)th hardening stiffness
$\lambda_1, \lambda_2, \lambda_3$	Principal stretches
$\mathbf{M}$	Mass
$\mathbf{m_x}, \mathbf{m_y}, \mathbf{m_z}$	Mass in element local directions
$\mathbf{M_x}, \mathbf{M_y}, \mathbf{M_z}$	Concentrated moments (local or global)
$\mathbf{M_x}, \mathbf{M_y}, \mathbf{M_z}, \mathbf{M_q}$	Flexural moments (local or global)
$\mathbf{M_{xy}}, \mathbf{M_{xz}}, \mathbf{M_{yz}}$	Torsional moments (local or global)
$\mathbf{M_1}, \mathbf{M_2}$	Concentrated loof moments (local or global)
μ	Coulomb friction coefficient
μ_{ri}, α_{ri}	Ogden rubber model constants
$\mathbf{N_x}, \mathbf{N_y}, \mathbf{N_z}, \mathbf{N_q}$	Membrane resultants (local or global)
$\mathbf{N_x}, \mathbf{N_y}, \mathbf{N_{xy}}$	Stress resultants
Nmax, Nmin	Principal stress resultants
$\mathbf{N_s}$	Maximum shear stress resultant
$\mathbf{N_e}$	Von Mises equivalent stress resultant
υ	Poisson's ratio
$\upsilon_{xy}, \upsilon_{xz}, \upsilon_{yz}$	Orthotropic Poisson's ratio
$\mathbf{P_x}, \mathbf{P_y}, \mathbf{P_z}$	Concentrated loads (global)
ρ	Mass density
$\mathbf{Q}$	Field loading for nodes
$\mathbf{q_a}$	Field face loading intensity
$\mathbf{q_v}$	Field volume loading intensity

q_x, q_y, q_z	Field fluxes (local or global)
q_s	Stress potential parameters
Sp	Plastic shear area
σ_y	Yield stress
σ_{yo}	Initial uniaxial yield stress
σ_x, σ_y, σ_z	Direct stresses (local or global)
σmax, σmin	Principal stresses
σ_{xy}, σ_{xz}, σ_{yz}	Shear stresses (local or global)
σs	Maximum shear stress
σ_e	Von Mises equivalent stress
T	Temperature
T, To	Final, initial temperatures
t1 ... tn	Nodal thicknesses
U, V, W	Displacements (global)
Φ	Field variable
Φe	External environmental temperature
ϕ	Frictional angle
ϕo	Initial frictional angle
ϕ	Body force potential
Vx, Vy, Vz	Nodal initial velocities (global)
$V1_1$, $V1_2$ ... $V3_3$	Left stretch tensor components
W_x, W_y, W_z	Uniformly distributed load intensities
X, Y, Z	Nodal coordinates (global)
Xcbf, Ycbf, Zcbf	Constant body forces (global)
Xo, Yo, Zo	Offsets of finite element model coordinate system from point about which global angular acceleration and velocities are applied
y_1, z_1 ... y_4, z_4	Cross sectional coordinates (local)
Zyp, Zzp	Torsional plastic modulus values
Zyyp, Zzzp	Flexural plastic modulus values
ω	Frequency of vibration
Ω_x, Ω_y, Ω_z	Angular velocities (global) (radians/sec)

Chapter 1 Introduction

Introduction

This manual contains details of the data sections available for input in a LUSAS datafile. The data sections required at any one time will depend upon the type of analysis to be carried out. Some of the sections are of specialised application, others are general to all problems. For example, the ELEMENT TOPOLOGY and NODE COORDINATES sections are required to define the problem in all analyses, however, the thermal conductivity of a material is only applicable to a heat transfer analysis. The specialised sections are referenced in the *Modeller User Manual* where a brief description is available of the problem type to which the command applies.

LUSAS Data Chapters

A full list of LUSAS data chapters, in the order in which they must be specified, is shown below.

Data section	Description
SYSTEM	System parameters
PROBLEM	Problem definition
UNITS	Unit definition
OPTION	User options
GROUP	Grouped data
ELEMENT TOPOLOGY	Element topology
SOLUTION ORDER	Equation solution order
SOLVER FRONTAL	Solve using Frontal method
SOLVER CONJUGATE GRADIENT	Solve using iterative Conjugate Gradient method
NODE COORDINATES	Node coordinates
LOCAL CARTESIAN COORDINATES	Local Cartesian coordinates
LOCAL CYLINDRICAL COORDINATES	Local cylindrical coordinates
LOCAL SPHERICAL COORDINATES	Local spherical coordinates

Data section	Description
GLOBAL CARTESIAN COORDINATES	Global Cartesian coordinates
SPACING	Spacing of nodes on a line
SPACING ARC RATIOS	Spacing of nodes on an arc
QUADRILATERAL SPACING	Spacing of nodes in a quadrilateral
COPY NODES	Copying nodes
RENUMBER NODES	Renumbering nodes
DELETE NODES	Deleting nodes
GEOMETRIC PROPERTIES	Geometric properties
COMPOSITE GEOMETRY	Laminated/composite shell/solid thicknesses
GEOMETRIC ASSIGNMENTS	Geometric property assignments
MATERIAL PROPERTIES	Isotropic model
MATERIAL PROPERTIES ORTHOTROPIC	Orthotropic plane stress model
MATERIAL PROPERTIES ORTHOTROPIC PLANE STRAIN	Orthotropic plane strain model
MATERIAL PROPERTIES ORTHOTROPIC THICK	Orthotropic thick model
MATERIAL PROPERTIES ORTHOTROPIC AXISYMMETRIC	Orthotropic axisymmetric model
MATERIAL PROPERTIES ORTHOTROPIC SOLID	Orthotropic solid model
MATERIAL PROPERTIES ANISOTROPIC	Anisotropic model
MATERIAL PROPERTIES ANISOTROPIC SOLID	Anisotropic model for solid elements
RIGIDITIES	Linear rigidity model
MATRIX PROPERTIES STIFFNESS	Stiffness matrix property definition
MATRIX PROPERTIES MASS	Mass matrix property definition
MATRIX PROPERTIES DAMPING	Damping matrix property definition
JOINT PROPERTIES	Standard linear joint model
JOINT PROPERTIES GENERAL	General linear joint model
JOINT PROPERTIES NONLINEAR 31	Standard elasto-plastic joint model
JOINT PROPERTIES NONLINEAR 32	General elasto-plastic joint model
JOINT PROPERTIES NONLINEAR 33	Nonlinear smooth contact model
JOINT PROPERTIES NONLINEAR 34	Nonlinear frictional contact model
JOINT PROPERTIES NONLINEAR 35	Nonlinear viscous damper model
MATERIAL PROPERTIES MASS	Mass for non-structural mass elements
PLASTIC DEFINITION	Nonlinear plasticity material data
MATERIAL PROPERTIES NONLINEAR 75	Optimised implicit Von Mises model
MATERIAL PROPERTIES NONLINEAR 64	Drucker-Prager yield surface model

Data section	Description
MATERIAL PROPERTIES NONLINEAR 61	Tresca yield surface model
MATERIAL PROPERTIES NONLINEAR 65	Non-Associated Mohr-Coulomb yield surface model
MATERIAL PROPERTIES NONLINEAR 94	Nonlinear multi-crack concrete model
MATERIAL PROPERTIES NONLINEAR 29	Stress resultant elasto-plastic model
MATERIAL PROPERTIES NONLINEAR USER	User defined nonlinear material model
MATERIAL PROPERTIES NONLINEAR RESULTANT USER	User defined nonlinear resultant material model
VISCOUS DEFINITION	Viscous material data
CREEP PROPERTIES	Creep material properties
CREEP PROPERTIES USER	User defined creep material properties
VISCOELASTIC PROPERTIES	Viscoelastic properties
VISCOELASTIC PROPERTIES USER	User defined viscoelastic properties
DAMAGE PROPERTIES	Damage properties
DAMAGE PROPERTIES USER	User defined damage model
DAMAGE PROPERTIES HASHIN	Hashin damage model for composites
TWO PHASE MATERIAL	Two phase material parameters
MATERIAL PROPERTIES RUBBER	Rubber material properties
MATERIAL PROPERTIES NONLINEAR 81	Volumetric crushing model
MATERIAL PROPERTIES NONLINEAR 86	CEB-FIP creep model
MATERIAL PROPERTIES NONLINEAR 89	Generic polymer material model
MATERIAL PROPERTIES NONLINEAR 27	2D elasto-plastic interface model
MATERIAL PROPERTIES NONLINEAR 26	3D elasto-plastic interface model
MATERIAL PROPERTIES NONLINEAR 25	Delamination interface properties
SHRINKAGE PROPERTIES CEB_FIP_90	Concrete shrinkage properties
SHRINKAGE PROPERTIES GENERAL	General shrinkage properties
SHRINKAGE PROPERTIES USER	User defined shrinkage model
MATERIAL PROPERTIES FIELD ISOTROPIC	Isotropic field model
MATERIAL PROPERTIES FIELD ORTHOTROPIC	Orthotropic field model

Data section	Description
MATERIAL PROPERTIES FIELD ORTHOTROPIC SOLID	Orthotropic solid field model
MATERIAL PROPERTIES FIELD LINK 18	Linear convection/radiation model
MATERIAL PROPERTIES FIELD LINK 19	Nonlinear convection/radiation model
COMPOSITE MATERIAL PROPERTIES	Laminated/composite shell/solid materials
MATERIAL ASSIGNMENTS	Material property assignments
COMPOSITE ASSIGNMENTS	Composite property assignments
ELEMENT AGES	Age of elements when using CEB-FIP concrete model
DEACTIVATE ELEMENTS	Deactivate elements
ACTIVATE ELEMENTS	Activate elements
DAMPING PROPERTIES	Frequency dependent Rayleigh parameters for viscous and/or structural damping
SLIDELINE PROPERTIES	Slideline property definition
SLIDELINE PROPERTIES USER	User defined slideline properties
SLIDELINE_SURFACE DEFINITION	Slideline surface definition
SLIDELINE ASSIGNMENTS	Slideline property assignments
THERMAL GAP PROPERTIES LINEAR	Linear thermal gap properties
THERMAL GAP PROPERTIES GENERAL	General thermal gap properties
THERMAL CONTACT PROPERTIES	Thermal contact properties
THERMAL RADIATION PROPERTIES	Thermal radiation properties
THERMAL ENVIRONMENT PROPERTIES	Thermal environment properties
THERMAL RADIATION SYMMETRY	Thermal radiation symmetry planes
THERMAL_SURFACE SYMMETRY ASSIGNMENT	Thermal radiation symmetry surface
THERMAL_SURFACE DEFINITION	Thermal surface definition
THERMAL_SURFACE PROPERTY ASSIGNMENT	Thermal surface property assignment
THERMAL ASSIGNMENT	Thermal gap and radiation surface assignment
VIEW_FACTOR OUTPUT	View factor output control
ENVIRONMENTAL NODE DEFINITION	Environmental node definition
ENVIRONMENTAL NODE ASSIGNMENTS	Environmental node assignment

Data section	Description
NODAL FREEDOMS	Number of freedoms at a node for thick shells
FREEDOM TEMPLATE	Optional definition of list of freedoms
CARTESIAN SETS	Local Cartesian sets
TRANSFORMED FREEDOMS	Transformed freedoms
CONSTRAINT EQUATIONS	Constraint equations
SUPPORT NODES	Support conditions
COUPLE	Coupled analyses initialisation
COUPLE READ	Coupled analysis data read
COUPLE WRITE	Coupled analysis data write
LOAD CASE	Load case definition
PDSP	Prescribed variables
CL	Concentrated loads
ELDS	Element loads
DLDL, DLDG	Distributed loads
DLEL, DLEG	Distributed element loads
PLDL, PLDG	Element point loads
UDL	Uniformly distributed loads
FLD	Face loads
CBF	Constant body forces
BFP	Body force potentials
BFPE	Element body force potentials
VELOCITY	Velocities
ACCELERATION	Acceleration
SSI	Initial stresses and strains at nodes
SSIE	Initial stresses and strains for elements
SSIG	Initial stresses and strains at Gauss points
SSR	Residual stresses at nodes
SSRE	Residual stresses for elements
SSRG	Residual stresses at Gauss points
TEMP	Temperature loads at nodes
TMPE	Temperature loads for elements
ENVT	Environmental temperatures
TEMPERATURE LOAD CASE	Temperature load case
TDET	Temperature dependent environmental temperatures
RIHG	Rate of internal heat generation
TEMPERATURE LOAD ASSIGNMENTS	Temperature load assignments
TDET	TDET assignments
RIHG	RIHG assignments
CURVE DEFINITION	General curve definitions
CURVE ASSIGNMENT	General curve assignment

Data section	Description
RETAINED FREEDOMS	Retained freedoms
ELEMENT OUTPUT	Element output control
NODE OUTPUT	Nodal output control
LOAD COMBINATION	Load combinations
ENVELOPE START	Enveloping results start
ENVELOPE FINISH	Enveloping results finish
SUPERELEMENT ASSIGN	Open a new superelement database
SUPERELEMENT DEFAULT	Change the default superelement database
SUPERELEMENT DEFINE	Defines a superelement
SUPERELEMENT USE	Uses the superelement in the residual structure
SUPERELEMENT RECOVER	Recovers superelement results
SUPERELEMENT DISTRIBUTE	Distributes mass and stiffness from the residual structure to the superelement
SUPERELEMENT MODAL_DATA	Utilises user modal data in the residual structure
NONLINEAR CONTROL	Nonlinear analysis control
DYNAMIC CONTROL	Dynamic analysis control
TRANSIENT CONTROL	Transient field analysis control
VISCOUS CONTROL	Creep analysis control
EIGENVALUE CONTROL	Eigenvalue analysis control
GUYAN CONTROL	Guyan reduced eigenvalue analysis
SPECTRAL CONTROL	Spectral response analysis control
HARMONIC_RESPONSE CONTROL	Harmonic Response analysis control
FOURIER CONTROL	Fourier analysis control
MODAL_DAMPING CONTROL	Control for distributed viscous and/or structural damping
PLOT FILE	Plot file generation
RESTART WRITE	Write to a restart file
RESTART READ	Read from a restart file
END	Problem termination.

Chapter 2 Syntax and Data Preparation

Reading and Writing LUSAS Data Syntax

Although the commands and numerical values required for each stage of the data input may vary for different analyses, there is a common form, or syntax, for every data line. This chapter defines the syntax rules to which every line of data input must conform.

LUSAS data syntax consists of command words and parameters. Command words instruct LUSAS of the data chapter or section being specified. Parameters provide details of the command section and are in the form of alphanumeric values. All data input for LUSAS is in a free format field, so reducing the time spent specifying data and reducing the possibility of errors. Certain command words or parameters are optional, and some data should be entered several times for different values. Occasionally, alternative command words or parameters can achieve the same objective.

How To Read Syntax Lines In This Manual

In this manual LUSAS data syntax lines are identified by a keyboard icon in the margin, and each separate line begins with a bullet as shown below. Enter the text as you read it on the page whilst following these rules:

- **Curved brackets** { } show alternative data input separated by a vertical line. You must choose one of the alternatives (mutually exclusive and compulsory). The brackets are not part of the syntax. Do not include them in your data input.
  ```
  {COMMAND 1 | COMMAND 2}
  ```

- **Square brackets** [ ] show optional data input. You may use one of the data input options, or none at all. The brackets are not part of the syntax. Do not include them in your data input.
  ```
  [option]
  ```

or

```
[option 1 | option 2]
```

- **Triangular brackets** < >i=1,n show data input that you should repeat along the same line according to the subscript. The chevrons and subscript are not part of the syntax. Do not include them in your data input.

An arrow is used to indicate that lines should be repeated for tabulated data entry. For example, the following syntax line means enter n values across and m down.

```
< Vi(1) >i=1,n
. . . .
< Vi(m) >i=1,n
```

Throughout this manual, the form of LUSAS input data is indicated by syntax lines as described above, and the parameters and command words used are defined beneath each group of syntax lines. The data input is divided into data chapters and data sections. Commonly a data chapter will contain several data sections. For example, the data chapter:

```
NONLINEAR CONTROL
```

can contain the data sections:

```
INCREMENTATION
STEP REDUCTION
ITERATION
BRACKETING
BRANCHING
CONVERGENCE
OUTPUT
INCREMENTAL COUPLE READ
INCREMENTAL COUPLE WRITE
TERMINATION
```

Data chapters must generally be specified in order. Within each data chapter, data sections may generally be specified freely. In either case, those chapters or sections which are not mandatory may be omitted. A full list of LUSAS data sections, and their usual order, is given in the Introduction.

Data Delimitation

The position of data items on a line is not important providing each word or parameter is sufficiently delimited. The delimiters used in LUSAS are blank spaces and commas.

Command Words

Command words instruct LUSAS of the data section currently being specified. Commonly a data section (or command line) will require the specification of more than one command word. The LUSAS command interpreter only decodes the first 4 letters of each word, hence long commands may be abbreviated.

Titles

For most data chapters in LUSAS it is possible to add a title which will be printed out as a heading in the output file. Whenever a syntax line indicates that a title may be written, it may consist of any characters required. For example:

```
NODE COORDINATES  [TITLE  title]
```

`title` descriptive text

Numerical Values

When a syntax line indicates that a numerical value is required, for convenience, it can be written in a number of ways as indicated in the syntax line below. There should be no spaces within a numerical value. Where an integer parameter is specifically indicated, integer input should be used. However, in general, integer and real numbers may be mixed freely. Small or large numerical values may be written in exponential (E) format.

```
[+ | -]< DIGIT >[.< DIGIT >][E[+ | -]< DIGIT >]
```

Simple Arithmetic Expressions

When a syntax line indicates that a numerical value is required it can be written as a simple arithmetic expression as indicated by the following:

```
VALUE < [*VALUE]  [/VALUE]  [**VALUE] >
```

The arithmetic expressions allowed in LUSAS are as follows, (note + and - are not allowed)

- Multiplication *
- Division /
- Exponentiation **

Some examples of the use of arithmetic expressions are shown below:

Expression	Means	Expression	Means
8*2	16	8**2	64
8/2/2	2	25E-1/2	1.25
8/2	4	8*2/4	4

25E-1*2 5 8/4**2 0.5

One of the uses of arithmetic expression input is that you can record in the data lines the component values which make up a numerical value. For example, the moment of inertia of a rectangular section is I=b*d*d*d/12 and you may wish to record the breadth b=4.3 and depth d=8.51, say, in the data as:

```
4.3*8.51**3/12
```

The arithmetic expression facility may also be employed for the conversion of units.

Comments

Comment lines may be placed anywhere in the data. A comment line must start with the letter C and must be followed by a space or comma. For example:

```
C  This is a comment
```

Alternatively, individual lines of data may be annotated using a colon (:). All information specified to the right-hand side of the colon will be treated as a comment. For example:

```
1 13 1 71E9 0.34      : This is a comment
```

Line Continuation

When the data input is too long for a line, it may be continued onto the next line by the use of the continuation indicator, which is a space followed by three full stops (...).

Data lines may be continued in this way onto as many lines as is required. However, a title may continue onto one extra line only.

Implied Sequence Generation

The implied sequence facility may be used as a simple method for the automatic generation of data, and is incorporated into many of the LUSAS data sections. The general form of the implied sequence generator is:

N N_{last} N_{diff}

where:

N is the first number in the sequence
N_{last} is the last number in the sequence
N_{diff} is the difference between consecutive sequence numbers

Each specified number must be an integer. LUSAS will check that a valid sequence is formed and will write an error message if this is not the case. Sequences comprising of one number only, for example 5, may be specified as (5 5 0) or (5 0 0).

The implied sequence generator is commonly used for the specification of sequences of nodes or elements, and its use can substantially reduce the amount of data specified. For example, the instruction to assign material number 3 to elements 10, 20 and 30 may be written simply as:

```
MATERIAL ASSIGNMENTS
10 30 10   3
```

Incremental Line Generation

Throughout all stages of LUSAS input there are commands which can be used to generate data lines.

FIRST $< V_i >_{i=1,n}$ **[R]**

INC $< \Delta_1\zeta_i >_{i=1,n}$ **[R_1]**

. . . . . .

INC $< \Delta_1\zeta_i >_{i=1,n}$ **[R_m]**

where:

Vi the numerical values on the data line to be generated. In certain circumstances these may consist of words which are simply duplicated.

N the number of numerical values on the data line

R the total number of repetitions of the first data line (inclusive) with unit increments added to each successive line. If this item is excluded a value of R=1 is assumed.

Δ1ζi the increments added to all data generated in the lower (m-1) levels. These increments may take any numerical values.

Rm the total number of repetitions of all data lines generated in the lower (m-1) levels.

The following simple examples illustrate the use of the general data generation. The data lines:

```
1  10 1  5.1
11 20 1  5.2
21 30 1  5.3
31 40 1  5.4
```

may be automatically generated using the data commands:

```
FIRST   1   10  1  5.1
  INC   10  10  0  0.1  4
```

When constructing element meshes by hand (as opposed to automatic mesh generation) it is often useful to number the nodes and elements such that the incremental line generator may be used. For example, when using 8 noded quadrilateral elements, the

node numbering may be specified so as to include fictitious central nodes. In this way a general element topology sequence is maintained.

TABLE Input

In certain circumstances, for example the specification of temperature dependent material properties, LUSAS requires a tabular form of input data. In such cases the TABLE data command may be used.

TABLE N [TITLE title]

< $V_{(1)i}$ >$_{i=1,n}$

. . . .

< $V_{(m)i}$ >$_{i=1,n}$

where:

N is the table identification number.
$V_{(k)i}$ are the values for the kth row of the table.
m is the number of rows in the table.

Notes

Data specified in tabulated form will be linearly interpolated at values between those values indicated.
Nonlinear variations may be input by increasing the number of tabulated data lines, and therefore approximating the nonlinear distribution by a number of straight lines.
The end of tabulated input is indicated by a new command word.

Chapter 3 LUSAS Data Input

System Parameters

The data section SYSTEM may be used to modify certain values which define particular machine and program parameters.

```
SYSTEM
vbname  =  n
END
```

vbname permitted system variable name (see below)
n the new value to be assigned to **vbname**

Notes

1. The SYSTEM header must be the first command in the input data.
2. SYSTEM input must always be terminated by the END command.

The following system variables may be modified (the default settings are shown in brackets):

BULKLF Linear viscosity coefficient for explicit dynamics elements (0.06)
BULKQF Quadratic viscosity coefficient for explicit dynamics elements (1.5)
CRKTOL Tolerance on ratio for cracking (0.01)
CTOL Creep strain convergence tolerance (1.0E-8)
DECAY Maximum diagonal decay allowed before problem termination (1.0E20)
DECAYL Limit of diagonal decay before warning messages are output (1.0E4)
EIGSCL Scaling factor used to compute the stresses from eigenvectors (1.0E-20)
HGVISC Hourglass viscosity coefficient for explicit dynamics elements (0.1)
LSMAX Maximum number of iterations for arc-length to compute load level (150)
LPGNUM The number of records of the paging file allowed in the cache (2500)
IECHO Echoes system status to screen (1)
IPGFAC Page factor used to set maximum size of files which can be created. IPFAC may be set to 1,2,4,8 (default),16,32 or 64. With IPGFAC=1 the

	max file size is 8Gb. IPGFAC =2 max file szie is 16Gb, IPGFAC=4 max file size is 32Gb etc.
MAXERR	Maximum number of times a single error will be output (50)
MAXNB	A large number which should not be exceeded by any element or node number (5000000)
MAXRSA	Maximum number of attempts allowed to set the symmetry tolerance **TOLSYM** (0)
MAXSEG	Maximum number of segments defining a radiation surface (100)
MEIGSH	Maximum number of eigen-solutions computed at a shift point (6)
MREFCE	Maximum number of elements sharing a common face (3)
MXEEST	Override estimate for total number of edges in structure (0)
MXELIM	Maximum number of equations to be eliminated simultaneously in the parallel frontal algorithm (6)
MXELGP	Maximum number of elements stored in an element group (Set automatically unless specified)
MXFEST	Override estimate for total number of faces in a structure (0)
MXIT	Maximum number of iterations in slideline slave search (500)
MXMNDZ	Maximum allowable frontwidth (5000)
MXSLAE	Maximum number of adjacent elements for contact nodes in slidelines or maximum number of adjacent thick shell elements at any node (64)
MXSLAN	Maximum number of adjacent nodes for contact nodes in slidelines (64)
MXWAIT	Maximum wait in seconds for DTF file before terminating analysis (900)
NCPU	Number of processors used in the parallel frontal algorithm (1)
NCITER	Iteration number at which to switch to iterative updates for multi crack concrete with crushing model (6)
NDPGSZ	Record length for any new direct access files in bytes (16384)
NIDX	Size of master index table (50000)
NLPZ	Number of real locations in the database (see on page)
NONNAT	Switch to ensure binary compatibility between PC and UNIX platforms (0, i.e., no binary compatibility)
NPGMAX	Maximum record length for direct access files in integer words (4096)
NPGS	Maximum number of records allowed in the cache (960)
PENTLY	Penalty stiffness coefficient used in constraint equations (0.0)
QMHDLM	Lower limit multiplier on hardening moduli for Mohr Coulomb model (0.01)
QTOL	Yield function convergence tolerance (1.0E-6)
SHLANG	Maximum angle (in degrees) between nodal normals after which 6 degrees of freedom are assigned to a node (maximum value=90). Applicable to thick shell elements. (20)
SLFNCS	Slideline normal force scale factor for contact cushioning (1.0)
SLSTCC	Slideline stiffness factor for close contact (1.0E-3)
SLSTFM	Slideline surface stiffness ratio (100.0)
SLSTPC	Slideline stiffness factor for pre-contact (1.0)

SSCALE Scaling parameter for creep algorithms (1.0)
STEFAN Stefan-Boltzmann constant, temperature units of the value input must be Kelvin (5.6697E-8 W/m2 K4)
STFINP In-plane stiffness parameter for flat shell elements (0.02)
STFSCL Stiffness matrix scaling factor for element deactivation (1.0E-6)
TOLFIJ Environment view factor tolerance, a radiation link may be formed if the environment view factor for a surface segment exceeds this value (-1.0)
TOLNOD Tolerance distance for node on node contact in a slideline analysis (1.0E-3)
TOLSYM Symmetry plane tolerance, a node is considered to lie in a symmetry plane if its perpendicular distance from the plane is less than this value (-1.0)
IZPPRB Compression level on PROBLEM database (default 0)
IZPSHP Compression level on SHAPES database (default 0)
IZPFRN Compression level on FRONTAL database (default 0)
IZPPLT Compression level on PLOT database (default 1)
IZPRST Compression level on RESTART database (default 1)
IZPDTF Compression level on DTF database (default 0)
IZPADP Compression level on ADAPT database (default 0)
IZPSUP Compression level on SUPER database (default 0)
IZPDBM Compression level on all other database (default 0)
BBOXF Scaling factor applied to increase bounding box which proscribes a slideline contact segment (default 1.2)

Problem Definition

The problem may be defined using the data sections for:

- Problem description (mandatory)
- Units definition (optional)
- Options (optional)

The PROBLEM data section is mandatory (specification of UNITS and OPTIONS is optional).

Problem

The PROBLEM data section defines the start of the problem data and is mandatory for all analysis types.

```
PROBLEM  [TITLE  title]
```

title Descriptive text

Notes

1. Data input for each problem must commence with the PROBLEM data section.
2. Title text can only be continued onto one additional line.

Units definition

LUSAS is unit independent, hence all data quantities must be specified in a consistent set of units (irrespective of whether the UNITS data section is specified). The UNITS data section may be used to name the units used, in order that the output quantities may be annotated.

```
UNITS  frc lth mas tim tem
```

frc	A word for the units of force
lth	A word for the units of length
mas	A word for the units of mass
tim	A word for the units of time
tem	A word for the units of temperature

Notes

3. All input data must be consistent with the units chosen for each problem. For dynamic analyses, if units are chosen for mass(m), length(l) and time(t), consistent units of force are given by,

   F = m a = ml/t/t

4. The UNITS command enables you to choose a symbol for force, length, mass, time and temperature, that will be printed out at the top of the column headings of the results output.
5. If a symbol for force, length or mass is specified which contains more than three characters, then only the first three characters will be output.

Options

The OPTIONS data section may be used to specify user definable analysis options.

OPTIONS < N_i >$_{i=1,n}$

N_i	The OPTION number(s)

Notes

1. For a problem requiring more than one option, repeat the line or add option numbers (negative option numbers may be used to disable previously enabled options), as in the following example:

   ```
   OPTION 40 45 55
   OPTION 87
   OPTION 77 -55
   ```

2. There is no limit to the number of OPTION lines in the data input.

3. Options may be specified and respecified at any point in the LUSAS datafile. Some options may be subsequently disabled by specifying a negative number; for example, the output of strains can be switched on and off at different points using Option 55, whilst it is not permissible to reset Option 87 which defines the problem type to be Total Lagrangian.
4. A full list of user options is included in the Appendices

Data Groups

Data groups may be used to collectively assign quantities to defined sequences of elements. GROUP is a general purpose utility which can be used to replace a series of first, last, difference element sequences within certain LUSAS data chapters (see *Notes*).

GROUP igroup [TITLE title]

L L_{last} L_{diff}

igroup	Group reference number
title	Descriptive text
L L_{last} L_{diff}	The first, last and difference between element numbers in a serie

Notes

1. The GROUP command must be specified before the ELEMENT TOPOLOGY data section.
2. The numbers defined in a group are checked to determine if a duplicate number has been specified.
3. The first and last number must be positive.
4. Groups may be utilised in GEOMETRIC ASSIGNMENTS, MATERIAL ASSIGNMENTS and COMPOSITE ASSIGNMENTS.

Element Topology

The data section ELEMENT TOPOLOGY is used to input the node numbers of the elements. The ELEMENT TOPOLOGY data section therefore describes the connectivity of the finite element discretisation.

type ELEMENT TOPOLOGY [TITLE title]

L < N_i >$_{i=1,n}$ < E_i >$_{i=1,m}$

type	The element type identifier as given in the description of each element; refer to the *LUSAS Element Reference Manual*.
L	The element number allocated to the particular element being defined.
N_i	The node numbers for each node of the particular element being defined.
E_i	The moment end conditions which apply only to BEAM, BRP2, BMS3, BTS3 and GRIL elements. Put **Ei** as R for a restrained rotation (default) and F for a free rotation. The BTS3 element also allows translational

degrees of freedom to be released (see the element descriptions in the *LUSAS Element Reference Manual* for further details).

n The total number of nodes for the particular element type.

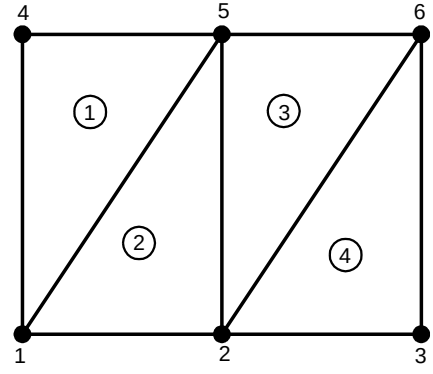
m The number of end releases.

Notes

1. For problems idealised with more than one element type, the header line ELEMENT TOPOLOGY is repeated for each element type and followed by the element number and node numbers for each element.
2. Each element must be given a unique identifying number. If an element number is repeated, the new element node numbers overwrite the previous element node numbers and an advisory message is printed out.
3. The elements should preferably be numbered in ascending order across the narrow direction of the structure (see Solution Order).
4. The element numbers may have omissions in the sequence and need not start at one. The order in which the element numbers are specified is arbitrary.
5. The element node numbers must be specified in the order shown in the element diagrams in the *LUSAS Element Reference Manual*.

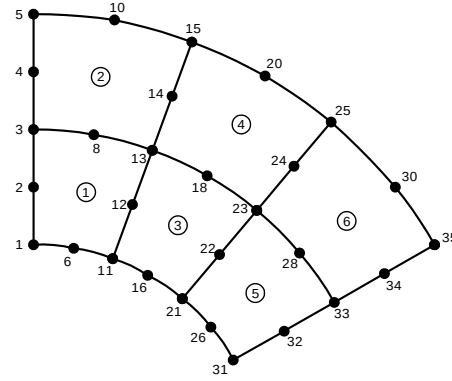
Example 1. Element Topology

```
TPM3 ELEMENT TOPOLOGY
  1    1  5  4
  2    1  2  5
  3    2  6  5
  4    2  3  6
```



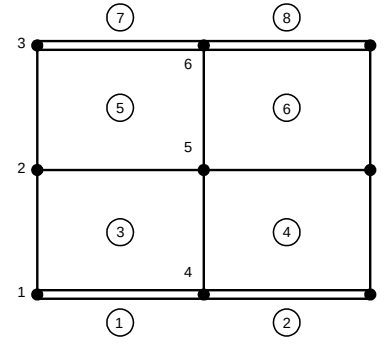
Example 2. Element Topology

```
QPM8 ELEMENT TOPOLOGY
FIRST  1 1  6  11 12
13 8  3  2
  INC  1 2  2  2  2  2
2  2  2  2
  INC  2 10 10 10 10
10 10 10 10 3
```



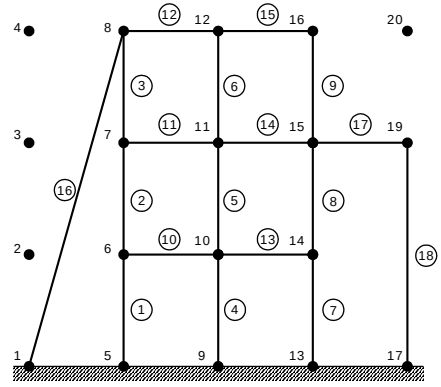
Example 3. Element Topology

```
QPM4 ELEMENT TOPOLOGY
FIRST  3 1 4 5 2
  INC  2 1 1 1 1  2
  INC  1 3 3 3 3  2
BAR2 ELEMENT TOPOLOGY
FIRST  1 1 4
  INC  1 3 3  2
  INC  6 2 2  2
```



Example 4. Element Topology

```
BEAM ELEMENT TOPOLOGY
       16 1  8    R F
FIRST  1  5  6    R R
  INC  1  1  1   (3)
  INC  3  4  4   (3)
FIRST  10 6  10   F R
  INC  1  1  1   (3)
FIRST  13 10 14   R R
  INC  1  1  1   (3)
       18 17 19   R R
       17 15 19   F R
```



Solution Order

The SOLUTION ORDER data section controls the efficient solution of the finite element discretisation. This command is not essential and may be omitted. However, in certain instances, a significant improvement in computation, both in time and cost may be achieved by the judicious selection of the element solution order.

```
SOLUTION ORDER  [ASCENDING | PRESENTED | AUTOMATIC
[nopt nitopt]]  [TITLE title]

{L Llast Ldiff | G  igroup}
```

nopt Automatic optimiser selection (default=4)
- =1 standard LUSAS optimiser
- =2 Akhras-Dhatt optimiser
- =3 Cuthill-McKee optimiser
- =4 Sloan optimiser

nitopt (when **nopt**=2) Number of optimising iterations (default=30)
(when **nopt**=3) Optimisation target (default=4)
- 1 RMS wavefront
- 2 Bandwidth
- 3 Profile
- 4 Max wavefront

L L_{last} L_{diff} The first, last and difference between element numbers in a series

G Command word which must be typed to use element groups.

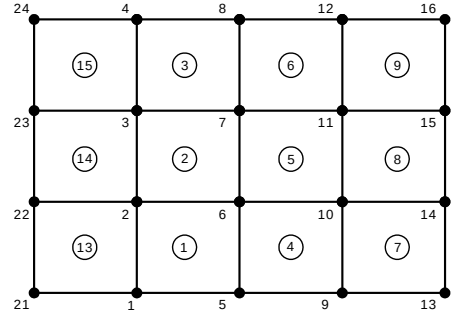
igroup Element group reference number.

Notes

1. If the header line is left out the default action is SOLUTION ORDER ASCENDING and the structure is solved according to ascending element number.
2. For SOLUTION ORDER PRESENTED the structure is solved according to the order in which the elements were presented in ELEMENT TOPOLOGY.
3. Element number data is not required for the SOLUTION ORDER PRESENTED.
4. With SOLUTION ORDER AUTOMATIC, Option 100 may be used to output the optimum element order for the frontal solution.
5. Each element number must only be specified once.
6. Fewer elements may be specified for SOLUTION ORDER than those specified in ELEMENT TOPOLOGY. If an element is not required in the solution it should be omitted from the data input.
7. Specification of element numbers not specified in ELEMENT TOPOLOGY is illegal.
8. The standard LUSAS optimiser should not be used for unconnected structures. If this occurs, an error message is output stating that the finite element mesh has an unconnected element. Option 100 will indicate all the elements within the structure as negative, those with a positive number are outside the structure.
9. Option 282 will switch the default optimiser to the standard LUSAS optimiser for compatibility with pre LUSAS version 12 data files.

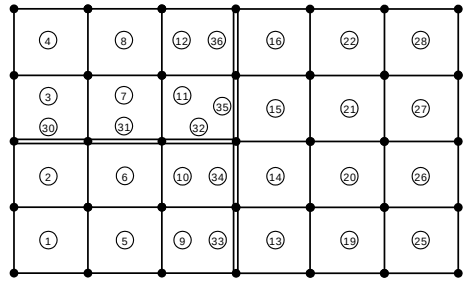
Example 1. Solution Order

```
QPM4 ELEMENT TOPOLOGY
FIRST  13 21 1  2  22
  INC  1  1  1  1  1  3
FIRST  1  1  5  6  2
  INC  1  1  1  1  1  3
  INC  3  4  4  4  4  3
SOLUTION ORDER PRESENTED
```



Example 2. Solution Order

```
SOLUTION ORDER
1  2  1
30
3  6  1
31
7  10 1
32
11 12 1
33 36 1
FIRST 13 16 1
INC 6  6  0  3
```



Solver Type

The SOLVER data section allows the specification of the solver to be used for the solution of the set of linear equations. This command is not essential and may be omitted, which will cause LUSAS to choose either the standard or the fast frontal solver, depending on the type of problem to be solved. However, in certain instances, a significant reduction in both computation time and memory may be achieved by choosing the iterative solver. It is also possible to assemble and write the global stiffness matrix and load vector(s) (or mass matrix, for eigenvalue problems) to binary files, without solving for the displacements.

```
SOLVER  {FRONTAL | FAST | ASSEMBLE}
```

or

```
SOLVER  CONJUGATE_GRADIENT {INCOMPLETE_CHOLESKY |
DECOUPLED | HIERARCHICAL}  [droptol]  [itmax]
```

droptol Drop tolerance parameter determining the size of the preconditioning matrix used during the conjugate gradient solution. The size of this matrix affects the nature of the iterative process, with larger preconditioning matrices giving rise to fewer, but more computationally expensive, iterations. The default value is 1.0, which produces relatively small matrices (i.e. fewer non-zero entries), which is suitable for well conditioned problems that do not require many iterations to achieve convergence. For more ill-conditioned problems, values in the range [1e-3, 1e-6] are recommended.

itmax Maximum number of conjugate gradient iterations to be processed (default = 5000).

Notes

1. The fast frontal solver will solve all problems except superelement analyses, Guyan reduction and non-linear problems using branching and bracketing.
2. For the fast frontal solver, the maximal and minimal pivots returned are based on magnitude, whereas the standard frontal solver returns pivots based on algebraic position. For example, if a problem gave rise to the three pivots 10.0, 0.1 and -1.0, the fast solver would return 10.0 and 0.1 as the maximum and minimum, respectively, whereas the standard solver would return 10.0 as the maximum and -1.0 as the minimum.
3. For the standard frontal solver, the concept of negative pivots is synonymous with that of negative *eigenvalues*, which signify when a bifurcation point has been reached during a non-linear analysis, and also whether a structure is loading or unloading. Thus warnings are given for negative pivots that are encountered during the solution phase. For the fast frontal solver, the concept of negative pivots is different from that of negative eigenvalues, and warnings of their existence are not given. For symmetric matrices, the number of negative eigenvalues is returned separately, and for non-symmetric matrices, the determinant of the stiffness matrix is returned, from which the *parity* of the eigenvalues (whether there are an even or odd number of negative eigenvalues) can be deduced. LUSAS uses this information during a non-linear analysis when using the fast solver, so the same results will be observed regardless of the solver used.
4. The conjugate gradient solver may *only* be used for linear, static analyses that give rise to symmetric, positive-definite stiffness matrices.
5. The INCOMPLETE_CHOLESKY option chooses Incomplete Cholesky preconditioning, which is applicable to all analyses for which the conjugate gradient solver may be used. With a judicious choice of drop tolerance, convergence is guaranteed for most problems.
6. The DECOUPLED option chooses Decoupled Incomplete Cholesky preconditioning, and may be used for all analyses except those involving tied slidelines, thermal surfaces and Fourier elements. It generally leads to faster overall solution times than Incomplete Cholesky preconditioning, although more iterations are required for convergence. For less well conditioned problems, the conjugate gradient algorithm may not converge using this technique, so care should be taken.
7. The HIERARCHICAL option chooses Hierarchical Decoupled Incomplete Cholesky preconditioning, which is only available for models consisting entirely of two- and three-dimensional, solid continuum, quadratic elements, and offers excellent convergence properties. It is by far the most effective technique for models of this type, and when used in conjunction with fine integration (OPTION 18) allows solutions to be obtained for relatively ill-conditioned problems. For very ill-conditioned problems of this type (e.g. where the average element aspect

ratio is high), an extra preconditioning option exists (OPTION 323) which will often yield a solution faster than using a direct solver.

8. When using the conjugate gradient solver with hierarchical basis preconditioning, if any midside degrees of freedom are supported or prescribed, their corresponding vertex neighbours must also be supported or prescribed. For example, if a midside node is fixed in the x-direction, all nodes on the same edge of that element must also be fixed (or prescribed) in the x-direction.
9. Problems involving constraint equations cannot currently be solved with the iterative solver, since the resulting stiffness matrix is non-positive-definite.
10. For problems with multiple load cases, iterative solvers are less efficient since a separate iterative process is required for each load case, and the total time taken will increase in proportion to the number of load cases. By contrast, direct solvers incur very little extra cost when solving for multiple load cases.
11. Guyan reduction and superelement analyses cannot be solved iteratively, since matrix reduction does not take place.
12. The iterative solver will perform very poorly if there is not enough main memory for the solution to proceed in-core. To guard against this, a data check (OPTION 51) may be performed (as with the direct solvers), which will estimate the amount of memory the iterative solver would use with the specified drop tolerance and choice of preconditioning technique.
13. If a convergence history of the iterative process is desired, OPTION 247 can be activated to write the residual norm of the solution vector to the output file after each iteration.
14. The preconditioning matrix can be stored using single precision storage (OPTION 248), which can significantly reduce the total amount of memory required for solution. For ill-conditioned problems, however, the rate of convergence may be affected, and the incomplete Cholesky factorisation may fail, hence this option must be used with care.
15. The iterative solver has limited error diagnostics to warn against ill-defined or incompletely specified models. If this is suspected, the analysis should be run through the standard frontal solver for more comprehensive error diagnostics.
16. For the iterative solver, the convergence criterion is a tolerance value of 1e-6 for the residual of the solution vector. If the solution returned by the iterative solver is deemed unsatisfactory, this tolerance can be lowered by altering the system parameter SOLTOL. It should not be raised under normal circumstances, unless an approximate solution only is required.
17. For the fast frontal and iterative solvers, the global matrix assembly involves the use of scratch files, since the size of the matrix data can be very large. For small problems which require many load increments or time steps, the global assembly process can be forced to remain in-core (OPTION 17) by placing it under the control of the data manager, assuming sufficient memory is available.
18. For the ASSEMBLE option, the data can be written to ASCII files by setting the system parameter MCHOUT to 0. The matrices are written in standard compressed

row (or Harwell-Boeing) format. For binary files, the order of the data written is as follows:

N	(integer - number of rows and columns)
NJA	(integer - number of non-zero entries in the matrix)
IA	(integer array, length N+1 - stores row pointers for the columns array JA)
JA	(integer array, length NJA - stores column indices for the non-zero entries)
A	(double precision array, length NJA - stores values for the non-zero entries)

Note that for symmetric matrices, only the upper triangular part of the matrix will be written to the file.

For vectors, the order of the data written to binary files is as follows:

N	(integer - total number of rows)
NVEC	(integer - number of load cases)
VECTOR	(double precision array, dimensioned (N, NVEC) - vector of values)

19. When using the fast frontal solver jobs sometimes fail because the presence of a large number of constraint equations causes an excessive amount of pivotting during the solution. To reduce the amount of pivotting a smaller value for the system parameter PVTTOL (default=0.01) can be specified in the data file.

Nodal Coordinates

The data chapter NODE COORDINATES is used to specify the nodal coordinates, defined in the global Cartesian system. All coordinates can be input using the NODE COORDINATE command. Alternatively the following facilities may be used within the NODE COORDINATES data chapter in order to aid or automate nodal coordinate generation:

- Local Coordinate Systems
- Spacing Nodal Coordinates on a Line
- Spacing Nodal Coordinates on an Arc
- Spacing Nodal Coordinates in a Quadrilateral
- Copying Nodes
- Renumbering Nodes
- Deleting Nodes

Node Coordinates

The NODE COORDINATES data section inputs the nodal coordinates in the global Cartesian coordinates of the problem.

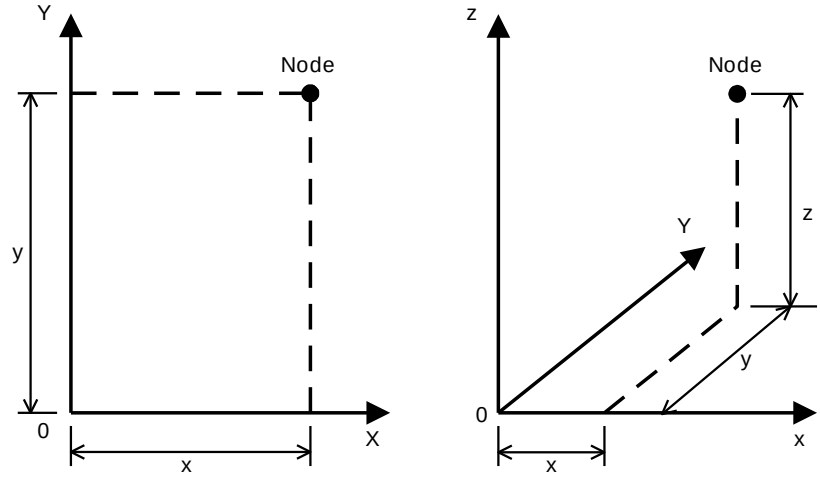
```
NODE COORDINATES  [TITLE  title]
N   X,Y [Z]
```

N The node number allocated to the particular node being defined.

X,Y,Z The global coordinates of the node. For 2D structures only X and Y need be specified.

Notes

1. If a node is repeated the new coordinate values overwrite the previous values and an advisory message is printed out.
2. LUSAS checks for nodes with same coordinates and if encountered, prints out an advisory message (Option 2 suppresses this check).
3. If several nodes are overwritten or specified in an arbitrary order, you can request output of the final node coordinates in ascending order with Option 30.
4. Extra dummy nodes, not associated with particular elements, may be specified. These dummy nodes can, for example, be used to simplify data generation.



2D and 3D Coordinate Definition

Local Coordinate Systems

Local Cartesian coordinates may be used to generate coordinate points in each of the local systems indicated that follow.

LOCAL {CARTESIAN | CYLINDRICAL | SPHERICAL}
COORDINATES N_0 N_x [N_{xy}]

N { X,Y | X,Y,Z | X,r,θ_x | r,θ_x,θ_c }

A return to global coordinates may be obtained following the command GLOBAL CARTESIAN COORDINATES.

GLOBAL CARTESIAN COORDINATES

No	The node defining position of local axis origin.
Nx	The node together with No defining the positive direction of local x-axis.
Nxy	The node defining the position of xy plane i.e. any point in the positive quadrant of the local xy plane. (Not required for 2D coordinates).
N	The node number allocated to the particular node being defined.
X, Y, Z	Local Cartesian coordinates (see Local Cartesian Coordinates).
X, r, θ_x	Local cylindrical coordinates (see Local Cylindrical Coordinates).
r, θ_x, θ_c	Local spherical coordinates (see Local Spherical Coordinates).

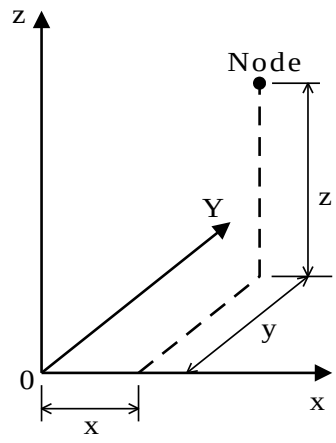
Notes

1. After insertion of a LOCAL COORDINATE header line, LUSAS assumes that all subsequent node coordinate data refers to that local coordinate system.
2. The coordinates of the nodes, with respect to the global coordinate axes, defining the position and orientation of the local axes must be defined prior to the insertion of LOCAL COORDINATE header line.
3. Other coordinate generation procedures such as FIRST, SPACING, QUADRILATERAL and COPY may be used in local coordinates (see subsequent commands in this section).
4. A set of LOCAL COORDINATE data must always be terminated by the GLOBAL CARTESIAN COORDINATE command. This command transforms the local node coordinates into the global coordinates in which LUSAS operates, and outputs the global coordinate values.
5. There is no limit to the number of sets of LOCAL COORDINATE data, but each set should be terminated by the GLOBAL CARTESIAN COORDINATE command before the subsequent set of LOCAL COORDINATE data is specified.
6. **θx** and **θc** are specified in degrees.

Local Cartesian Coordinates

For LOCAL CARTESIAN COORDINATES the following definitions apply:

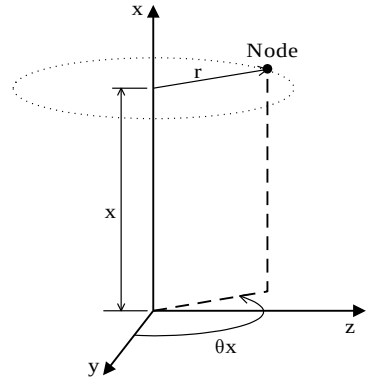
x	distance from the local origin in the local x-direction.
y	distance from the local origin in the local y-direction.
z	distance from the local origin in the local z-direction.



Local Cylindrical Coordinates

For LOCAL CYLINDRICAL COORDINATES the following definitions apply:

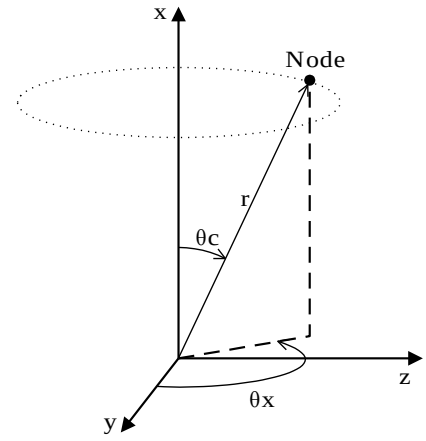
x	distance from the local origin in the local x-direction.
r	radius from the local x-axis in the local yz-plane.
qx	angle in degrees from the side of the local xy-plane about the local x-axis (right-hand screw rule).



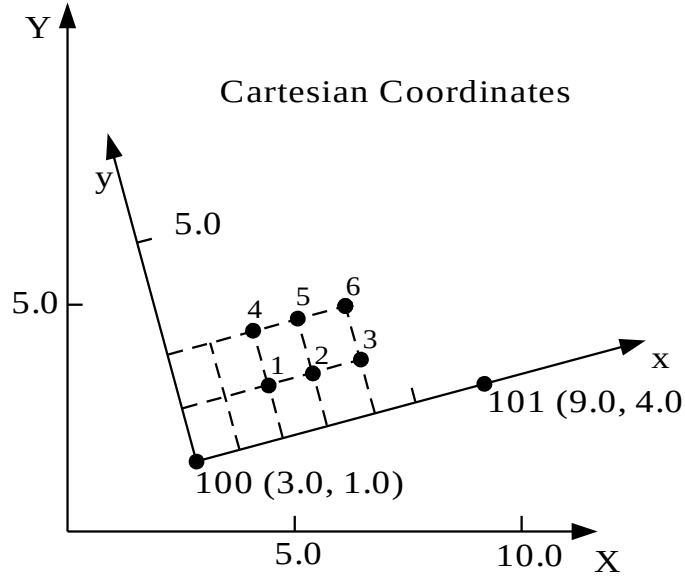
Local Spherical Coordinates

For LOCAL SPHERICAL COORDINATES the following definitions apply:

r	radius from the local origin
qx	angle in degrees from the positive quadrant of the local xy-plane about the local x-axis. The rotation is clockwise when looking along the x-axis in the positive direction (right hand screw rule).
qc	angle in degrees from the local x-axis to the radius line.



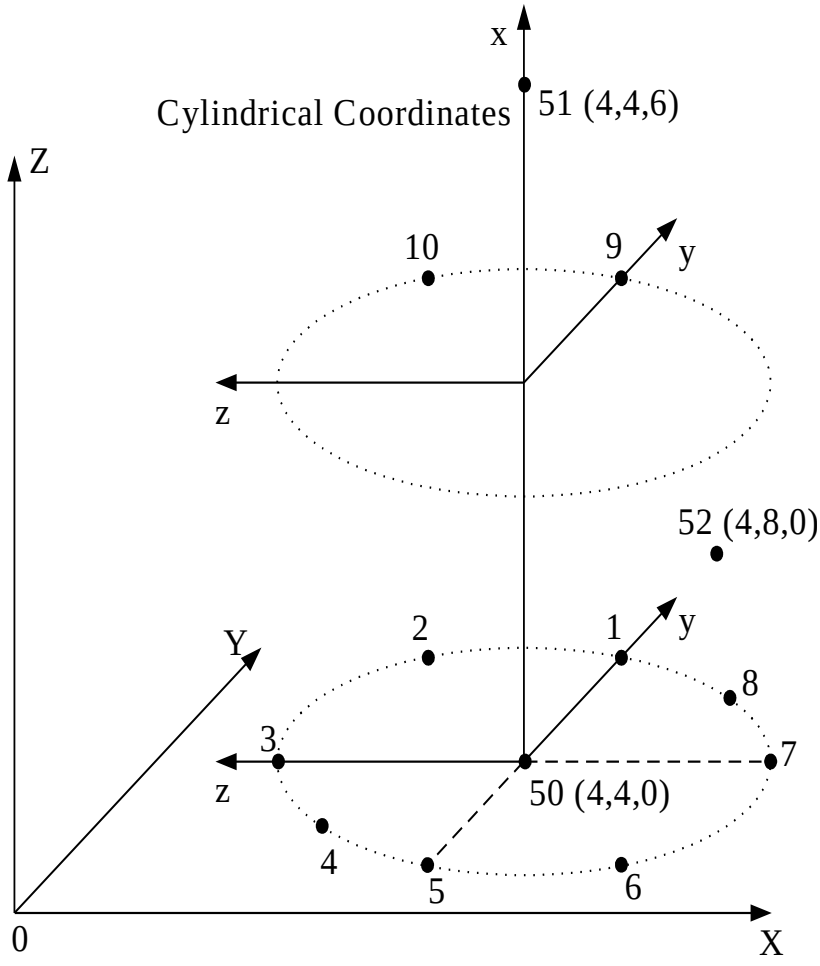
Example 1. Local Cartesian Coordinates




```
NODE COORDINATES
  100  3.0  1.0
  101  9.0  4.0
LOCAL CARTESIAN COORDINATES 100 101
```

```
FIRST  1  2  1
  INC  1  1  0  3
  INC  3  0  1  2
GLOBAL CARTESIAN COORDINATES
```

Example 2. Local Cylindrical Coordinates




```
NODE COORDINATES
  50  4  4  0
  51  4  4  6
```

```
  52  4  8  0
LOCAL CYLINDRICAL COORDINATES 50 51 52
  1   0  2  0
  2   0  2  45
  3   0  2  90
  4   0  2  135
  5   0  2  180
  6   0  2  225
  7   0  2  270
  8   0  2  315
  9   3  2  0
  10  3  2  45
GLOBAL CARTESIAN COORDINATES
```

Spacing Nodal Coordinates on a Line

The SPACING command computes the coordinates of a line of nodes from the coordinates of the end nodes and the defined spacing ratios.

SPACING [{X | Y | Z} RATIOS]

N N_{last} N_{diff} < S_i >$_{i=1,n}$

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes to be spaced.

S_i The ratios of the spaces between consecutive nodes. For M spaces with the same value S use an asterisk to automatically repeat value as M*S. Note that **i** ≤ 120 even when M*S format is used.

n The number of spaces between consecutive nodes.

Notes

1. The coordinates of the first and last nodes of a line must be specified before the spacing data line.
2. Projections of the line connecting the N and Nlast nodes on the X, Y and Z axes may be evaluated by using the SPACING X RATIOS, SPACING Y RATIOS and SPACING Z RATIOS commands respectively.
3. It is permissible to specify up to 120 spacing ratios. If the data will not fit on to one line use the LUSAS line continuation symbol (three dots …) to continue onto a second line.
4. Additional spacing data may be specified without repeating the header line.

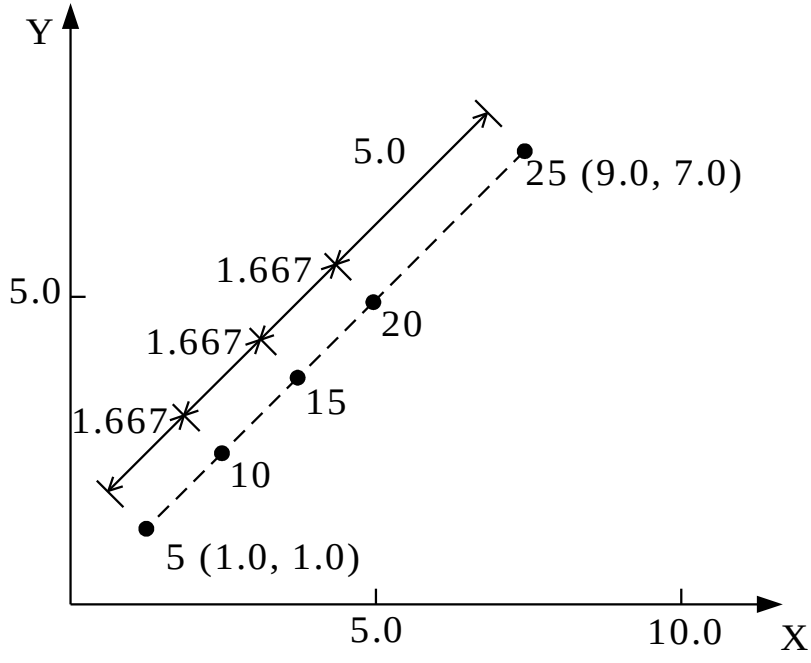
Example 1. Node Coordinates Spacing on a Line

This data file segment:

```
NODE COORDINATES
  5   1  1
  25  9  7
SPACING
  5  25 5   1.667 1.667
1.667  5.0
```

Has the same effect as this one:

```
NODE COORDINATES
  5   1  1
  25  9  7
SPACING
  5  25 5   3*1.667  5.0
```



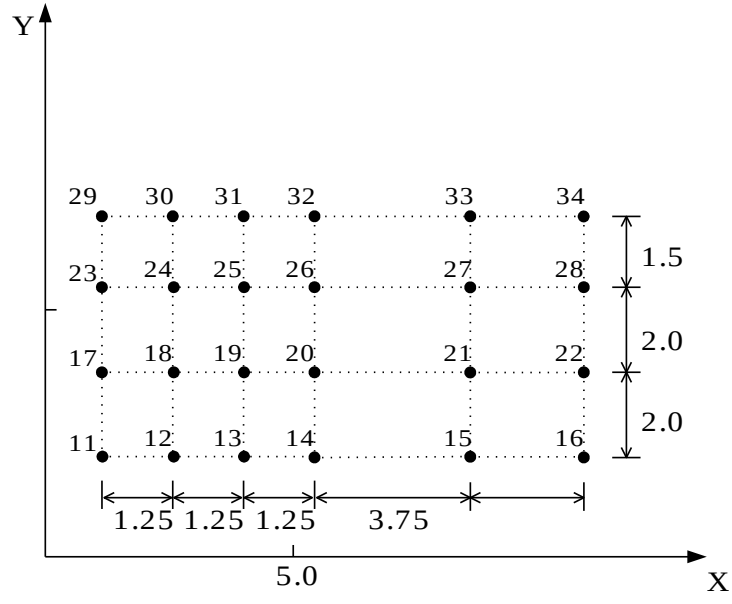
Example 2. Node Coordinates Spacing on a Line

This data file segment:

```
NODE COORDINATES
  11  1.0   2.0
  16  11.0  2.0
  29  1.0   7.5
  34  11.0  7.5
SPACING
  11  29  6  2*2.0   1.5
  16  34  6  2*2.0   1.5
  11  16  1  3*1.25
3.75  2.5
  17  22  1  3*1.25
3.75  2.5
  23  28  1  3*1.25
3.75  2.5
  29  34  1  3*1.25
3.75  2.5
```

Has the same effect as this one:

```
NODE COORDINATES
11 1.0 2.0
16 11.0 2.0
29 1.0 7.5
34 11.0 7.5
SPACING
FIRST 11 29 6 2*2.0 1.5
INC 5 5 (2)
FIRST 11 16 1 3*1.25
3.75 2.5
INC 6 6 (4)
```



Spacing Nodal Coordinates on an Arc

The SPACING ARC RATIOS data section computes the coordinates of a circular line of nodes from the coordinates of the end nodes and the defined arc ratios.

SPACING [X | Y] ARC RATIOS X_c Y_c [Z_c] [N_d]

N Nlast Ndiff < Si >i=1,n

Xc, Yc, Zc The coordinates of the centre of the circle.

Nd The node defining the direction in which nodes are to be spaced around circle.

N Nlast Ndiff The first node, last node and difference between nodes of the series of nodes to be spaced.

Si The ratio of the arc spaces between consecutive nodes. For M spaces with the same value S use an asterisk to automatically repeat value as M*S. Note that **i** ≤ 120 even when M*S format is used.

n The number of arc spaces between consecutive nodes.

Notes

1. The coordinates of the first, last and centre nodes of a circular line must be specified before the spacing data line. The radius between the centre node and the **N** and **Nlast** nodes must be equal.
2. It is permissible to specify up to 120 arc spacing ratios. If the data will not fit onto one line, use the LUSAS line continuation symbol (three dots …) to continue onto a second line.
3. Additional spacing data may be specified without repeating the header line.
4. The program will assume that the nodes will be spaced around the shortest arc length between the first and last nodes. For sweep angles greater than 180°, a node, lying in the plane and direction of the circular line to be generated, must be specified on the header line. The coordinates of this direction node must be specified prior to the header line.
5. Incremental generation may be used to generate several circular lines with the same centre. For M arc spaces with the same increment value S, use the asterisk repeat facility as M*S. Zero arc spacing ratio increments need not be specified if the total number of lines are put in brackets.
6. The projection of the X or Y coordinate of the arc onto the X or Y axis is calculated using the SPACING X ARC RATIOS and SPACING Y ARC RATIOS respectively. The projected nodes lie in the same Z-plane as the centre of the circle.
7. When using SPACING X (or Y) ARC RATIOS the first and last Y (or X) coordinates must not change sign over the segment of arc being generated.

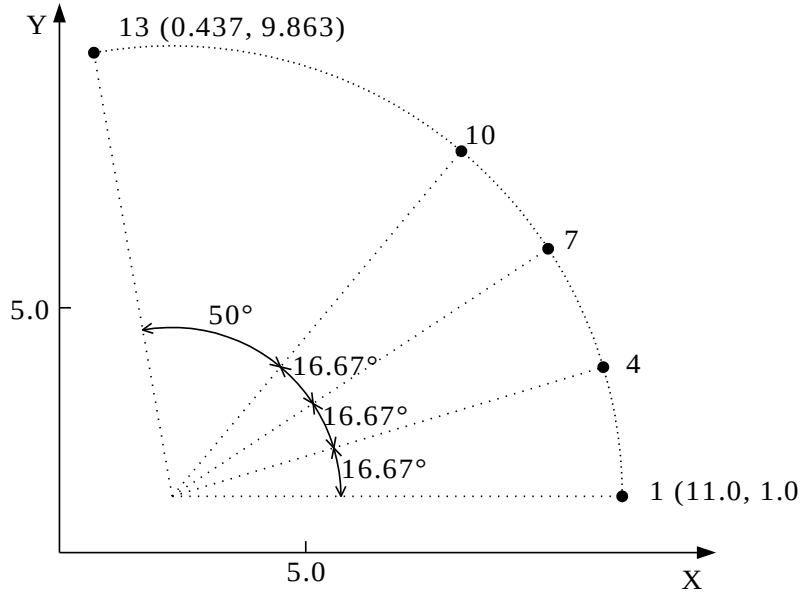
Example 1. Node Coordinates Spacing on an Arc

This data file segment:

```
NODE COORDINATES
  1   11.0   1.0
  13  0.437  9.863
SPACING ARC RATIOS 2.0
1.0
  1  13  3   16.67
16.67  16.67  50.0
```

Has the same effect as this:

```
NODE COORDINATES
  1   11.0   1.0
  13  0.437  9.863
SPACING ARC RATIOS 2.0
1.0
  1  13  3   3*16.67
50.0
```



Example 2. Node Coordinates Spacing on an Arc

This data file segment:

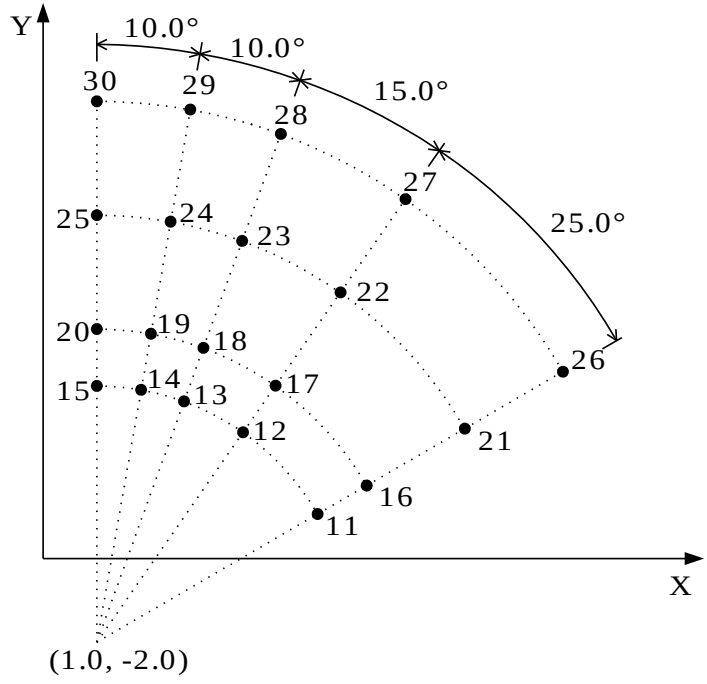
```
NODE COORDINATES
  11  5.33  0.5
  26  9.66  3.0
```

Has the same effect as this:

```
NODE COORDINATES
  11  5.33  0.5
  26  9.66  3.0
```

```
  15  1.0   3.0
  30  1.0   8.0
SPACING
  15  30  5   1.0  2.0
2.0
  11  26  5   1.0  2.0
2.0
SPACING ARC RATIOS 1.0 -
2.0
  11  15  1  25  15
2*10
  16  20  1  25  15
2*10
  21  25  1  25  15
2*10
```

```
  15  1.0   3.0
  30  1.0   8.0
SPACING
  15  30  5   1.0  2.0
2.0
  11  26  5   1.0  2.0
2.0
SPACING ARC RATIOS 1.0 -
2.0
FIRST  11 15 1  25  15
2*10
  INC  5  5  0 (4)
```



Spacing Nodal Coordinates in a Quadrilateral

The data section QUADRILATERAL SPACING generates node coordinates for plane or parabolic quadrilateral zones.

```
QUADRILATERAL SPACING
N  { X,Y | X,Y,Z }
SIDE POINTS
N Nlast Ndiff [ Xs,Ys | Xs,Ys,Zs ]
```

N A corner node number of the quadrilateral zone to be generated.

X, Y, Z The global coordinates of a corner node. For 2D structures, only X, Y are specified. Four corner node data lines are required in any order.

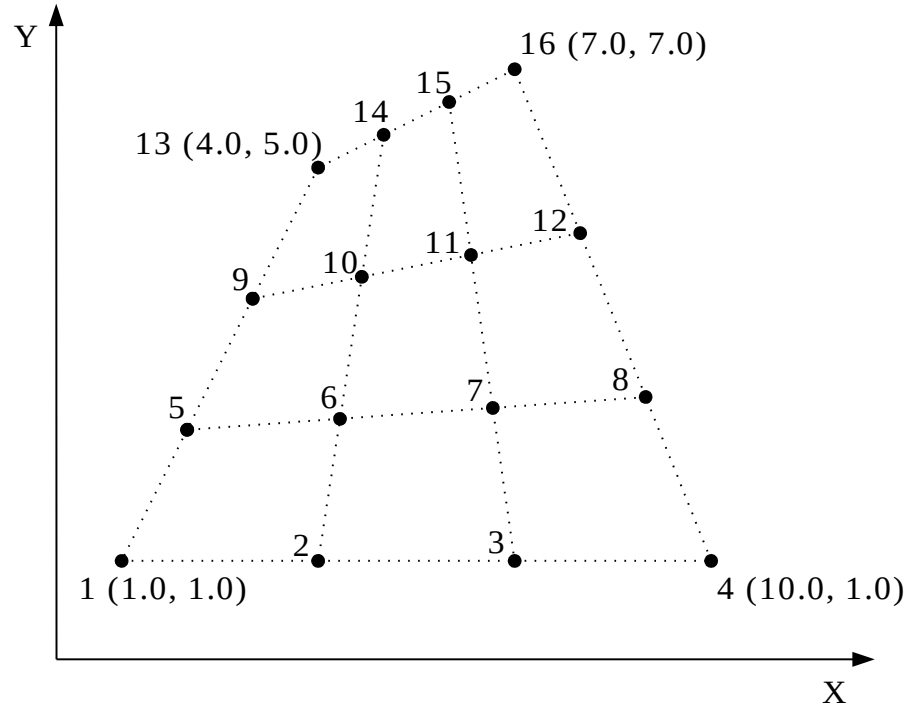
N Nlast Ndiff The first node, last node and difference between nodes of the series of nodes along a side of the quadrilateral zone.

Xs, Ys, Zs The global coordinates of a point along a side of the quadrilateral zone which defines the parabolic shape and grading of the line of nodes. This point must lie inside the central half of the side and need not be coincident with any node. Four side point data lines are required in any order.

Notes

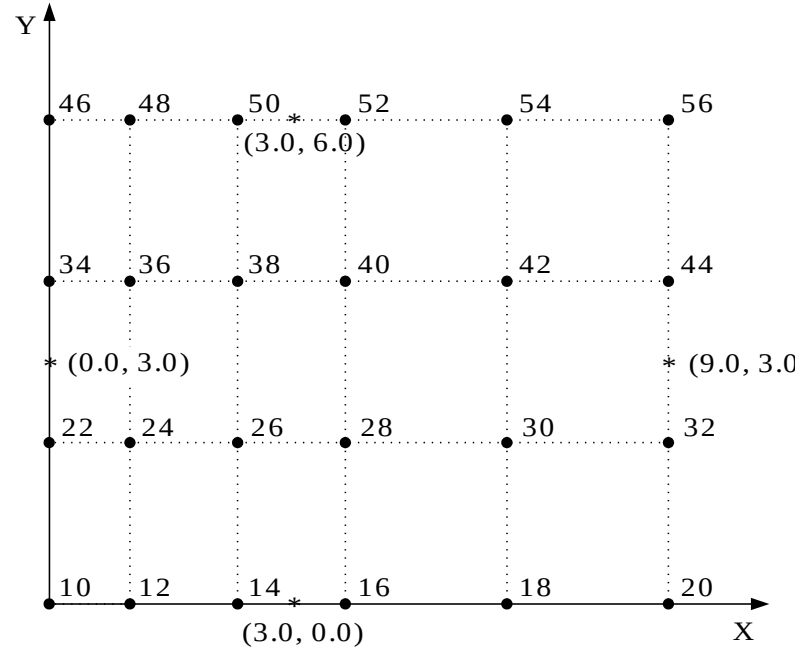
1. If the coordinates of the side points are omitted a straight sided regularly spaced quadrilateral will be generated.
2. The angle subtended at any corner of a quadrilateral zone must be less than 180° otherwise non-uniqueness of mapping may result.

Example 1. Node Coordinates Quadrilateral Spacing




```
NODE COORDINATES
QUADRILATERAL SPACING
  8   2   1
  11  7   1
  29  8   5
  26  2.5 5
SIDE POINTS
  8   11  1   4.5  1.6
  11  29  6   7    3
  26  29  1   5.25 5.6
  8   26  6   1.8  3
```

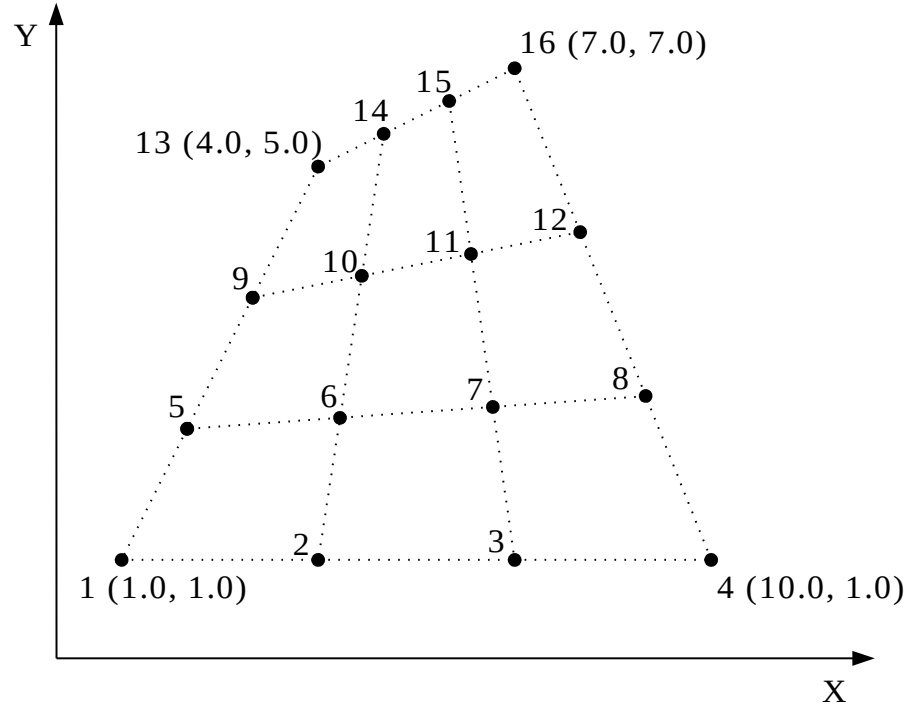
Example 2. Node Coordinates Quadrilateral Spacing




```
NODE COORDINATES
QUADRILATERAL SPACING
  10  0   0
  20  9   0
  46  0   6
  56  9   6
SIDE POINTS
  10  20  2   3   0
  20  56  12  9   3
  46  56  2   3   6
  10  46  12  0   3
```

The number of nodes on either side of the mid-node are equal, but the spacing of the nodes will only be equal if the mid-node happens to bisect the side.

Example 3. Node Coordinates Quadrilateral Spacing




```
NODE COORDINATES
QUADRILATERAL SPACING
  1   1   1
  4   10  1
  13  4   5
  16  7   7
SIDE POINTS
  1   4   1
  1   13  4
  13  16  1
  4   16  4
```

Copying Nodes

The data section COPY NODES copies the coordinates of a series of nodes to another series of nodes.

COPY NODES N1 $N1_{last}$ $N1_{diff}$ TO N2 $N2_{last}$ $N2_{diff}$

N1 N1last N1diff The first node, last node and difference between nodes of the series of node coordinates to be copied.

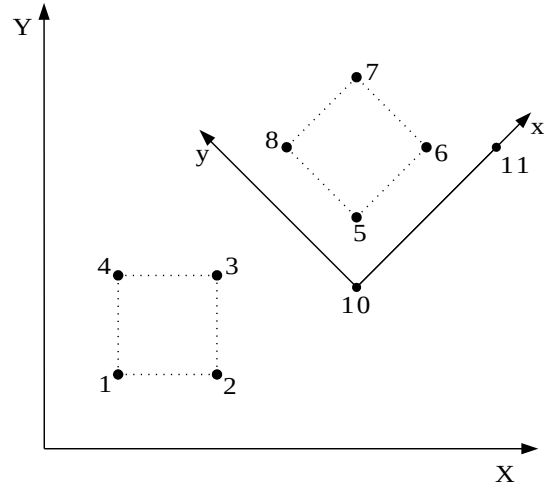
N2 N2last N2diff The first node, last node and difference between nodes of the recipient series of nodes.

Notes

The coordinates of the first series of nodes must be specified prior to the use of this command.

1. Any node coordinates in the second series of nodes which were previously specified will be overwritten.
2. The number of nodes in both series must be equal.
3. This command could, for example, be used in LOCAL COORDINATES to copy a repetitive pattern of nodes to a new position and orientation.

Example. Node Coordinates Copying Nodes




```
NODE COORDINATES
  1   1  1
  2   2  1
```

```
    3   2  2
    4   1  2
    10  4  2
    11  6  3
  LOCAL CARTESIAN COORDINATES 10 11
  COPY NODES  1 4 1  TO  5 8 1
  GLOBAL CARTESIAN COORDINATES
```

Renumbering Nodes

The data section RENUMBER NODES renumbers a series of node numbers with a new series of node numbers.

RENUMBER NODES N1 $N1_{last}$ $N1_{diff}$ TO N2 $N2_{last}$ $N2_{diff}$

N1 N1last N1diff The first node, last node and difference between nodes of the series of node coordinates to be renumbered.

N2 N2last N2diff The first node, last node and difference between nodes of the recipient series of nodes.

Notes

1. The node coordinates of the series of nodes to be renumbered must be specified prior to the use of this command.
2. Any node coordinates of the series of new nodes which were previously specified will be overwritten.
3. The number of nodes in both series must be equal.

Deleting Nodes

The data section DELETE NODES deletes a series of node coordinates from the LUSAS database.

DELETE NODES N N_{last} N_{diff}

N Nlast Ndiff The first node, last node and difference between nodes of the series of node coordinates to be deleted.

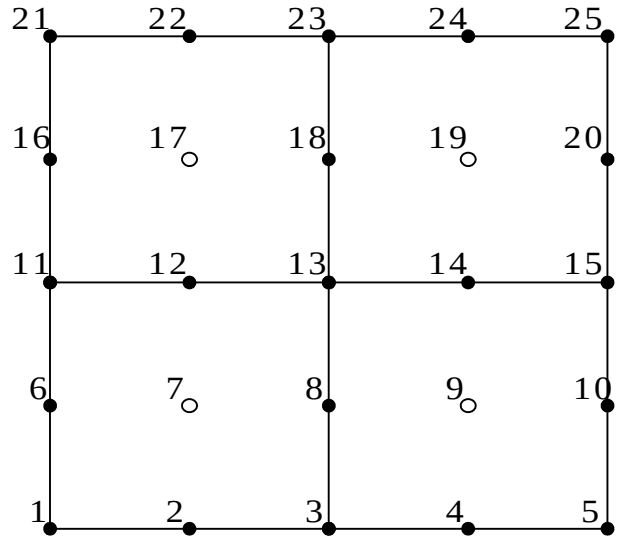
Note

1. If a node for deletion has not been previously specified, a warning message will be output.

Example. Node Coordinates Deleting Nodes

```
  DELETE NODES 7 9 2
```

```
DELETE NODES 17 19 2
```



Geometric Properties

The data section GEOMETRIC PROPERTIES is used to define the geometric property values for the specified element type. Not all elements will require the input of geometric properties; for example, the geometric properties for a membrane element will be the element thickness at each node, whilst there is no equivalent property for the solid elements. The *LUSAS Element Reference Manual* should be consulted for geometric property details of each element type.

GEOMETRIC PROPERTIES are assigned to a series of elements using the GEOMETRIC ASSIGNMENTS data section.

type GEOMETRIC PROPERTIES [CONSTANT] [nxs] [TITLE title]

igmp < G_i >$_{i=1,n}$

type The element type identifier as given in the description of each element. Refer to the *LUSAS Element Reference Manual*.

CONSTANT Specifies that the GEOMETRIC PROPERTIES are the same for all nodes on the element (see *Notes*).

nxs The number of quadrilateral cross-sections defining the total cross-section (used for beam elements, see *Notes*).

igmp The geometric property reference number (see Geometric Assignments).

Gi The geometric property values for the element type specified, see element descriptions in the *LUSAS Element Reference Manual f*or definition of values.

n Number of geometric properties to be input.

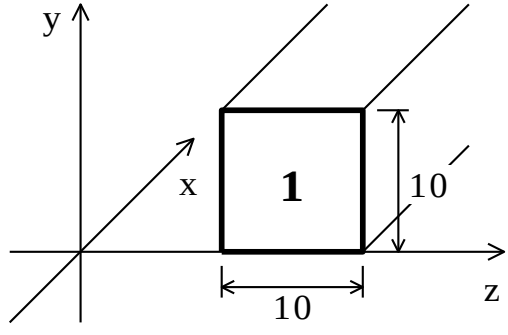
Notes

1. Some element types in LUSAS do not require geometric property input, in which case this section should be omitted.
2. If an element is repeated, the new geometric properties overwrite the previous values for that element and an informative message is printed in the output file.
3. When the CONSTANT parameter is used, the geometric properties for only 1 node need be defined and the others are assumed to be the same.
4. The parameter nxs can only be utilised with beam elements that require quadrilateral cross-sections to be defined via the local coordinates: BMX3 element in 2D; BXS4 and BXL4 elements in 3D. The maximum number allowed for parameter nxs is 2000 (approximately 200 cross section geometries).
5. For compatibility with previous versions of LUSAS, prior to LUSAS 12, geometric properties may still be associated with elements directly using the element first, last, inc facility. To use this approach Option -117 must now be set. The CONSTANT and nxs parameters cannot be utilised with this approach.

Examples of Geometric Properties

Input for 3D problem comprising thin beam elements with and without quadrilateral cross-sections (see *LUSAS Element Reference Manual* for definition of values):

Example 1. Geometric Properties Without Quadrilateral Cross-Sections



Properties defined for each node:

```
BS3 GEOMETRIC PROPERTIES
```

```
11  100.0 833.33  1666.66  0.0  0.0  0.0 ...
    100.0 833.33  1666.66  0.0  0.0  0.0 ...
    100.0 833.33  1666.66  0.0  0.0  0.0
```

Properties defined constant for all nodes:

```
BS3 GEOMETRIC PROPERTIES   CONSTANT
12  100.0  833.33 1666.66  0.0  0.0  0.0
```

Example 2. Geometric Properties With Quadrilateral Cross-Sections

Cross section defined at each node. Input four y,z pairs at each node followed by the number of integration points in the local y and z directions respectively.

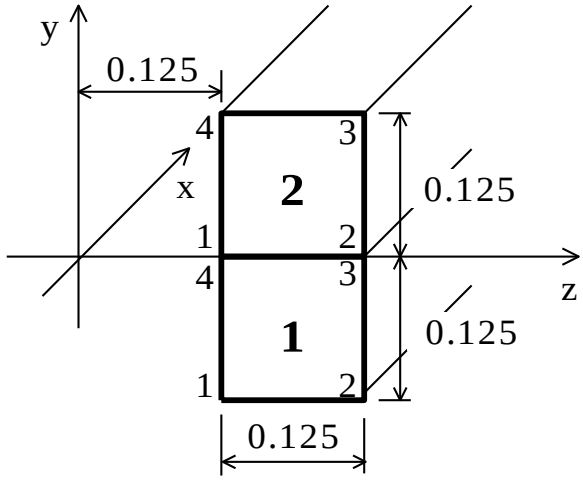
```
BXL4 GEOMETRIC PROPERTIES
21  0.25 -0.125  0.5 -0.125  0.5 0.125  0.25 0.125
    0.25 -0.125  0.5 -0.125  0.5 0.125  0.25 0.125
    0.25 -0.125  0.5 -0.125  0.5 0.125  0.25 0.125 3 8
```

Cross section defined constant for all nodes. Input four y,z pairs followed by the number of integration points in the local y and z directions respectively.

```
BXL4 GEOMETRIC PROPERTIES   CONSTANT
21  0.25 -0.125  0.5 -0.125  0.5 0.125  0.25 0.125 3 8
```

Cross section defined as two rectangles constant for all nodes. Input four y,z pairs followed by the number of integration points in the local y and z directions respectively for each quadrilateral in the section.

```
BXL4 GEOMETRIC PROPERTIES   CONSTANT    2
21  0.25 -0.125  0.5 -0.125  0.5 0.0   0.25 0.0   3 4
    0.25  0.0    0.5  0.0    0.5 0.125 0.25 0.125 3 4
```



Composite Geometry

The data section COMPOSITE GEOMETRY defines the thicknesses of layers used to laminate a composite material. The number of layers defined in this data section must be the same as the number of layers used in the accompanying COMPOSITE MATERIAL section. The data is input in tabular form where rows relate to layers and columns to element nodes. The lay-up sequence is always defined sequentially from the lower to upper surface of the element. COMPOSITE GEOMETRY and COMPOSITE MATERIALS are assigned to elements through the COMPOSITE ASSIGNMENT data section.

COMPOSITE GEOMETRY [TITLE title]

TABLE icgp

$\mathbf{t1}_1$ **[<** $\mathbf{t1}_i$ **>**$_{\mathbf{i=2,nnode}}$**]**

..

$\mathbf{tnlayr}_1$ **[<** $\mathbf{tnlayr}_i$ **>**$_{\mathbf{i=2,nnode}}$**]**

icgp Composite geometry set number.
tji Thickness of layer **j** at node **i** (see *Notes*).
nlayr Total number of layers.
nnode Number of element nodes.

Notes

1. Node order is defined by element topology. If the layer thickness is the same at each node then only the thickness at node 1 need be defined.

2. The layer thickness may be specified as a ratio of the total thickness defined under GEOMETRIC PROPERTIES for semiloof shells or of the depth defined by the element topology for composite solids.

Geometric Assignments

The data section GEOMETRIC ASSIGNMENTS is used to assign defined geometric property sets to single, groups or sequences of elements.

```
GEOMETRIC ASSIGNMENTS  [TITLE title]
{L Llast Ldiff | G  igroup}  igmp  [igmpv]
```

L L_{last} L_{diff} The first, last and difference between elements with the same geometric assignment.

igmp The geometric property reference number (see Geometric Properties)

igmpv The varying geometric property reference number if it is defined in the LUSAS Modeller pre-processing model. This number is saved in the LUSAS Modeller results file for use in post-processing.

G A command word which must be typed to use element groups.

igroup The element group reference number (see Defining Data Groups).

Example. Geometric Assignments

Nodal thicknesses for a single curved shell element (QSL8) with 8 nodes:

```
QSL8 GEOMETRIC PROPERTIES
  2  2.5  2.5 2.5  3.0  3.5  3.5  3.5  3.0
GEOMETRIC ASSIGNMENTS
  1 0 0    2
```

Material Properties

Every element declared in the model discretisation must be assigned a material property. Material property definitions may be classified into one of the following groups:

- **Linear Material Properties**
- **Matrix Properties**
- **Joint Material Properties**
- **Mass Properties**
- **Nonlinear Material Properties**
- **Field Material Properties**
- **Composite Material Properties**

Each set of data specified under MATERIAL PROPERTIES must have a unique material identification number associated with it. This allows a group of elements to be assigned a set of material properties under MATERIAL ASSIGNMENTS.

Material properties specified under MATERIAL PROPERTIES can be combined with the PLASTIC DEFINITION, VISCOUS DEFINITION and/or DAMAGE PROPERTIES definitions via the MATERIAL ASSIGNMENTS data chapter.

Temperature dependent material properties may be input for both field and structural elements. In this case the TABLE command must directly follow the particular material properties command. Lines of data listing the material properties at particular reference temperatures are then input.

The following restrictions apply to the use of the temperature dependent material properties:

- Limited to continuum models (von Mises, Tresca, Mohr-Coulomb, Drucker-Prager), i.e., not available for stress resultant model
- Limited to formulations based on total strains (geometric linearity and Total Lagrangian or Co-rotational geometric nonlinearity)
- Hardening modulus values are not temperature dependent.

Notes

1. Superfluous properties or rigidities for elements not present in a structure may be specified.
2. For a more detailed description of each constitutive model refer to the *LUSAS Theory Manual*.
3. The *LUSAS Element Reference Manual* defines the material properties that are applicable for each of the element types.
4. For compatibility with previous versions of LUSAS, material properties may still be associated with elements directly using the element first/last/inc facility. To use this approach Option -118 and/or -146 must be set.

Linear Material Properties

The following linear elastic material models are available:

- **Isotropic**
- **Orthotropic Plane Stress**
- **Orthotropic Plane Strain**
- **Orthotropic Thick**
- **Orthotropic Axisymmetric**
- **Orthotropic Solid**

- **Anisotropic**
- **Rigidity specification**

Linear Isotropic Model

The data section MATERIAL PROPERTIES is used to define the material properties for linear elastic isotropic materials.

```
MATERIAL PROPERTIES  [TITLE  title]

imat  E ν [ρ α ar br T]
```

imat	The material property identification number
E	Young's modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
T	Reference temperature

Linear Orthotropic Plane Stress Model

The data section MATERIAL PROPERTIES ORTHOTROPIC is used to define the material properties for linear orthotropic plane stress materials.

```
MATERIAL PROPERTIES ORTHOTROPIC

imat  Ex Ey Gxy νxy [θ ρ αx αy αxy ar br T]
```

imat	The material property identification number
Ex,Ey	Young's modulus values
Gxy	Shear modulus
νxy	Poisson's ratio
θ	Angle of orthotropy in degrees relative to the reference axis (see *Notes*).
ρ	Mass density
αx, αy, αxy	Coefficients of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
T	Reference temperature

Notes

1. Subscripts refer to the element reference axes, where reference axes may be local or global (see *Local Axes* in the *LUSAS Element Reference Manual f*or the proposed element type). If q (about z) is set to zero, the reference axes are used for defining material properties.

2. When using MATERIAL PROPERTIES ORTHOTROPIC care must be taken to ensure that all properties are input to sufficient numerical accuracy. Failure to do this may result in erroneous answers.

Linear Orthotropic Plane Strain Model

The data section MATERIAL PROPERTIES ORTHOTROPIC PLANE STRAIN is used to define the material properties for linear orthotropic plane strain materials.

```
MATERIAL PROPERTIES ORTHOTROPIC PLANE STRAIN
imat  Ex Ey Ez Gxy νxy νyz νxz [θ ρ αx αy αxy αz ar br T]
```

imat The material property identification number
Ex,Ey,Ez Young's modulus values
Gxy Shear modulus
νxy,νyz,νxz Poisson's ratios
θ Angle of orthotropy in degrees relative to the reference axis (see *Notes*).
ρ Mass density
αx,αy,αxy,αz Coefficients of thermal expansion
ar Mass Rayleigh damping constant
br Stiffness Rayleigh damping constant
T Reference temperature

Notes

1. Subscripts refer to the element reference axes, where reference axes may be local or global (see *Local Axes* in the *LUSAS Element Reference Manual* for the proposed element type). If q (about z) is set to zero, the reference axes are used for defining material properties.
2. When using MATERIAL PROPERTIES ORTHOTROPIC care must be taken to ensure that all properties are input to sufficient numerical accuracy. Failure to do this may result in erroneous answers.

Linear Orthotropic Thick Model

The data section MATERIAL PROPERTIES ORTHOTROPIC THICK is used to define the material properties for linear orthotropic thick materials.

```
MATERIAL PROPERTIES ORTHOTROPIC THICK
imat  Ex Ey Gxy νxy Gxz Gyz [θ ρ αx αy αxy αxz αyz ar br T]
```

imat The material property identification number
Ex,Ey Young's modulus values
Gxy,Gxz,Gyz Shear modulus values
νxy Poisson's ratio

θ Angle of orthotropy in degrees relative to the reference axis (see *Notes*).
ρ Mass density
αx,αy,
αxy,αxz,αyz Coefficients of thermal expansion
ar Mass Rayleigh damping constant
br Stiffness Rayleigh damping constant
T Reference temperature

Notes

1. Subscripts refer to the element reference axes, where reference axes may be local or global (see *Local Axes* in the *LUSAS Element Reference Manual* for the proposed element type). If q (about z) is set to zero, the reference axes are used for defining material properties.
2. When using MATERIAL PROPERTIES ORTHOTROPIC care must be taken to ensure that all properties are input to sufficient numerical accuracy. Failure to do this may result in erroneous answers.

Linear Orthotropic Axisymmetric Model

The data section MATERIAL PROPERTIES ORTHOTROPIC AXISYMMETRIC is used to define the material properties for linear orthotropic axisymmetric materials.

MATERIAL PROPERTIES ORTHOTROPIC AXISYMMETRIC

imat E_x E_y E_z G_{xy} ν_{xy} ν_{yz} ν_{xz} [θ ρ α_x α_y α_{xy} α_z a_r b_r T]

imat The material property identification number
Ex,Ey,Ez Young's modulus values
Gxy Shear modulus
νxy,νyz,νxz Poisson's ratios
θ Angle of orthotropy in degrees relative to the reference axis (see *Notes*).
ρ Mass density
αx,αy,αxy,αz Coefficients of thermal expansion
ar Mass Rayleigh damping constant
br Stiffness Rayleigh damping constant
T Reference temperature

Notes

1. Subscripts refer to the element reference axes, where reference axes may be local or global (see *Local Axes* in the *LUSAS Element Reference Manual* for the proposed element type). If q (about z) is set to zero, the reference axes are used for defining material properties.

2. When using MATERIAL PROPERTIES ORTHOTROPIC care must be taken to ensure that all properties are input to sufficient numerical accuracy. Failure to do this may result in erroneous answers.

Linear Orthotropic Solid Model

The data section MATERIAL PROPERTIES ORTHOTROPIC SOLID is used to define the material properties for linear orthotropic solid materials.

MATERIAL PROPERTIES ORTHOTROPIC SOLID

imat nset E_x E_y E_z G_{xy} G_{yz} G_{xz} ν_{xy} ν_{yz} ν_{xz} [ρ α_x α_y α_z α_{xy} α_{yz} α_{xz} a_r b_r] T

imat The material property identification number
nset The CARTESIAN SET number used to define the local axis directions.
Ex,Ey,Ez Young's modulus values
Gxy,Gyz,Gxz Shear modulus values
νxy,νyz,νxz Poisson's ratios
ρ Mass density
αx,αy,αz,
αxy,αyz,αxz Coefficients of thermal expansion
ar Mass Rayleigh damping constant
br Stiffness Rayleigh damping constant
T Reference temperature

Notes

1. Subscripts refer to the element reference axes, where reference axes may be local or global (see *Local Axes* in the *LUSAS Element Reference Manual* for the proposed element type). For the solid model, the orthotropy is defined by the CARTESIAN SET command. If **nset** is set to zero, the orthotropy coincides with the reference axes.
2. When using MATERIAL PROPERTIES ORTHOTROPIC care must be taken to ensure that all properties are input to sufficient numerical accuracy. Failure to do this may result in erroneous answers.

Linear Anisotropic Model

The data section MATERIAL PROPERTIES ANISOTROPIC is used to define arbitrary constitutive equations relating stress to strain. The material modulus matrix is input on a component by component basis. Note that symmetry is assumed so that only the upper triangle of the matrix is required. The matrix is defined column by column.

MATERIAL PROPERTIES ANISOTROPIC {n | SOLID}

imat {ρ a_r b_r T θ|nset ρ a_r b_r T} < α_i >$_{i=1,n}$ < D_i >$_{i=1,n(n+1)/2}$

imat	The material property identification number
ρ	Mass density
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
T	Reference temperature
θ	Angle of anisotropy (degrees) relative to the element reference axes (see *Notes*)
nset	The CARTESIAN SET number used to define the axes of anisotropy (required for ANISOTROPIC SOLID only).
αi	Coefficients of thermal expansion
Di	Values in upper triangular half of modulus matrix.
n	The number of stress/strain components for element (=6 for SOLID)

Notes

1. The element reference axes may be local or global (see *Local Axes* in the *LUSAS Element Reference Manual f*or the proposed element type). If **nset** or q is set to zero, the anisotropy coincides with the reference axes.
2. The upper triangle components of the modulus matrix only are entered (the components are entered column by column), and:

   - **Di** The **m** components of the upper triangle of the modulus matrix.
   - **m** The number of components of the modulus matrix, where **m** may be computed from **m**=**n**(**n**+1)/2

Linear Rigidity Model

The data section RIGIDITIES is used to define the in-plane and bending rigidities from prior explicit integration through the element thickness.

RIGIDITIES n

imat ρ a_r b_r T θ < α_i >$_{i=1,n}$ < D_i >$_{i=1,n(n+1)/2}$

imat	The material property identification number
n	The number of stress/strain resultants for the element
ρ	Mass density
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
T	Reference temperature
θ	Angle of orthotropy relative to the reference axis (degrees)
αi	Coefficients of thermal expansion
Di	The values in the upper triangular half of the rigidity matrix

Note

1. The element reference axes may be local or global (see *Local Axes* in the *LUSAS Element Reference Manual* for the proposed element type). If q is set to zero, the anisotropy coincides with the reference axes.

Example 1. Membrane Behaviour

```
RIGIDITIES 3
imat, ρ, a_r, b_r, T, q, α_1 ... α_3, D_1 ... D_6
```

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} D_1 & D_2 & D_4 \\ D_2 & D_3 & D_5 \\ D_4 & D_5 & D_6 \end{bmatrix} \left\{ \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{Bmatrix} - \begin{Bmatrix} \varepsilon_{xo} \\ \varepsilon_{yo} \\ \varepsilon_{xyo} \end{Bmatrix} \right\} + \begin{Bmatrix} N_{xo} \\ N_{yo} \\ N_{xyo} \end{Bmatrix}$$

where:

N are the membrane stress resultants (force per unit width).
D membrane rigidities.
e membrane strains.

and for isotropic behaviour, where t is the thickness:

$$D_1 = D_3 = \frac{Et}{1-\nu^2} \qquad D_2 = \frac{\nu Et}{1-\nu^2} \qquad D_6 = \frac{Et}{2(1+\nu)}$$
$$D_4 = D_5 = 0$$

The initial strains due to a temperature rise T are:

$$\varepsilon_{ot} = \begin{Bmatrix} \varepsilon_{xo} \\ \varepsilon_{yo} \\ \varepsilon_{xyo} \end{Bmatrix} + T \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix}$$

Example 2. Thin Plate Flexural Behaviour

```
RIGIDITIES 3
imat, ρ, a_r, b_r, T, q, α_1 ... α_3, D_1 ... D_6
```

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_1 & D_2 & D_4 \\ D_2 & D_3 & D_5 \\ D_4 & D_5 & D_6 \end{bmatrix} \left\{ \begin{Bmatrix} \Psi_x \\ \Psi_y \\ \Psi_{xy} \end{Bmatrix} - \begin{Bmatrix} \Psi_{xo} \\ \Psi_{yo} \\ \Psi_{xyo} \end{Bmatrix} \right\} + \begin{Bmatrix} M_{xo} \\ M_{yo} \\ M_{xyo} \end{Bmatrix}$$

where:

M are the flexural stress resultants (moments per unit width).
D flexural rigidities.
Y flexural strains given by:

$$\begin{Bmatrix} \Psi_x \\ \Psi_y \\ \Psi_{xy} \end{Bmatrix} = \begin{Bmatrix} -\dfrac{\partial^2 w}{\partial x^2} \\ -\dfrac{\partial^2 w}{\partial y^2} \\ -\dfrac{2\partial^2 w}{\partial x \partial y} \end{Bmatrix}$$

and for an isotropic plate for example, where t is the thickness:

$$D_1 = D_3 = \frac{Et^3}{12(1-\nu^2)} \quad D_2 = \frac{\nu Et^3}{12(1-\nu^2)} \quad D_6 = \frac{Et^3}{24(1+\nu)} \quad D4$$

$= D5 = 0$

The initial strains due to a temperature rise T are:

$$\Psi_{ot} = \begin{Bmatrix} \Psi_{xo} \\ \Psi_{yo} \\ \Psi_{xyo} \end{Bmatrix} = \frac{\partial T}{\partial z} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix}$$

Example 3. Thick Plate Flexural Behaviour

```
RIGIDITIES 5

imat, ρ, a1, b1, T, q, α1 ... α5, D1 ... D15
```

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \\ S_{xz} \\ S_{yz} \end{Bmatrix} = \begin{bmatrix} D_1 & D_2 & D_4 & D_7 & D_{11} \\ D_2 & D_3 & D_5 & D_8 & D_{12} \\ D_4 & D_5 & D_6 & D_9 & D_{13} \\ D_7 & D_8 & D_9 & D_{10} & D_{14} \\ D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{bmatrix} \left\{ \begin{Bmatrix} \Psi_x \\ \Psi_y \\ \Psi_{xy} \\ \Gamma_{xz} \\ \Gamma_{yz} \end{Bmatrix} - \begin{Bmatrix} \Psi_{xo} \\ \Psi_{yo} \\ \Psi_{xyo} \\ \Gamma_{xzo} \\ \Gamma_{yzo} \end{Bmatrix} \right\} + \begin{Bmatrix} M_{xo} \\ M_{yo} \\ M_{xyo} \\ S_{xzo} \\ S_{yzo} \end{Bmatrix}$$

where:

M are the flexural stress resultants (moments per unit width).

S shear stress resultants (shear force per unit width).
D flexural and shear rigidities.
Y flexural strains given by:

$$\begin{Bmatrix} \Psi_x \\ \Psi_y \\ \Psi_{xy} \end{Bmatrix} = \begin{Bmatrix} -\dfrac{\partial^2 w}{\partial x^2} \\ -\dfrac{\partial^2 w}{\partial y^2} \\ -\dfrac{2\partial^2 w}{\partial x \partial y} \end{Bmatrix}$$

G shear strains given by:

$$\begin{Bmatrix} \Gamma xz \\ \Gamma yz \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial w}{\partial x} + \dfrac{\partial u}{\partial z} \\ \dfrac{\partial w}{\partial y} + \dfrac{\partial v}{\partial z} \end{Bmatrix}$$

and for an isotropic plate for example:

$$D_{10} = D_{15} = \frac{Et}{2(1+\nu)k}$$

$$D_7 = D_8 = D_9 = D_{11} = D_{12} = D_{13} = D_{14} = 0$$

where t is the plate thickness and k is a factor taken as 1.2 which provides the correct shear strain energy when the shear strain is assumed constant through the plate thickness. D1 to D6 are the same as defined for the thin plate flexural behaviour (see Example 2. Thin Plate Flexural Behaviour).

The initial strains due to a temperature rise T are:

$$\begin{Bmatrix} \Psi_{ot} \\ \varepsilon_{ot} \end{Bmatrix} = \begin{Bmatrix} \Psi_{xo} \\ \Psi_{yo} \\ \Psi_{xyo} \\ \Gamma_{xzo} \\ \Gamma_{yzo} \end{Bmatrix} = \frac{\partial T}{\partial z} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 0 \\ 0 \end{Bmatrix} + T \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \alpha_4 \\ \alpha_5 \end{Bmatrix}$$

Example 4. Shell Behaviour

```
RIGIDITIES 6
```

imat, ρ, a_r, b_r, T, q, α_1 ... α_6, D_1 ... D_{21}

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_1 & D_2 & D_4 & D_7 & D_{11} & D_{16} \\ D_2 & D_3 & D_5 & D_8 & D_{12} & D_{17} \\ D_4 & D_5 & D_6 & D_9 & D_{13} & D_{18} \\ D_7 & D_8 & D_9 & D_{10} & D_{14} & D_{19} \\ D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{20} \\ D_{16} & D_{17} & D_{18} & D_{19} & D_{20} & D_{21} \end{bmatrix} \left\{ \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \Gamma_{xy} \\ \Psi_x \\ \Psi_y \\ \Psi_{xy} \end{Bmatrix} - \begin{Bmatrix} \varepsilon_{xo} \\ \varepsilon_{yo} \\ \Gamma_{xyo} \\ \Psi_{xo} \\ \Psi_{yo} \\ \Psi_{xyo} \end{Bmatrix} \right\} + \begin{Bmatrix} N_{xo} \\ N_{yo} \\ N_{xyo} \\ M_{xo} \\ M_{yo} \\ M_{xyo} \end{Bmatrix}$$

where:

N	are the membrane stress resultants (forces per unit width).
M	are the flexural stress resultants (moments per unit width).
D	flexural and shear rigidities.
e	membrane strains.
G	flexural strains.

The initial strains due to a temperature rise T are:

$$\begin{Bmatrix} \Psi_{ot} \\ \varepsilon_{ot} \end{Bmatrix} = \begin{Bmatrix} \varepsilon_{xo} \\ \varepsilon_{yo} \\ \Gamma_{xyo} \\ \Psi_{xo} \\ \Psi_{yo} \\ \Psi_{xyo} \end{Bmatrix} - T \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 0 \\ 0 \\ 0 \end{Bmatrix} + \frac{dt}{dz} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{Bmatrix}$$

Matrix Properties

The data section MATRIX PROPERTIES is used to explicitly define the linear properties of the stiffness, mass and damping matrices for joint elements in the local coordinate system.

MATRIX PROPERTIES {STIFFNESS | MASS | DAMPING} [N]

imat {< K_i >$_{i=1,m}$ | < M_i >$_{i=1,m}$ | < C_i >$_{i=1,m}$}

imat	The material property identification number
Ki	The values in upper triangular half of local element stiffness matrix.
Mi	The values in upper triangular half of local element mass matrix.
Ci	The values in upper triangular half of local element damping matrix.

N Size of matrix (total number of freedoms for element in question). The default value is given by twice the maximum number of freedoms at a node for the structure in question.

m Number of components of the matrix where m is calculated from **m**=**N**(**N**+1)/2

Notes

1. Either the stiffness, mass or damping matrix can be specified for an element. If more than one matrix is specified the last matrix overwrites all previous matrices. To specify more than one matrix property overlay the number of elements required.
2. The force/displacement relationship is defined as:

$$P = T^T K T a$$

where:

P The global nodal forces

T A transformation matrix relating local freedoms to global

K The local element stiffness matrix stored in upper triangular form.

a The global element freedoms (displacements).

Joint Material Properties

Joint material models are used to define the material properties for linear and nonlinear joint models. Ten types of joint model are available:

- **Standard Linear Joint Model**
- **General Linear Joint Model**
- **Standard Nonlinear Elasto-Plastic Joint Model**
- **General Nonlinear Elasto-Plastic Joint Model**
- **Nonlinear Joint Model For Smooth Contact**
- **Nonlinear Joint Model For Frictional Contact**
- **Visco-elastic Dampers**
  - **Kelvin**
  - **Four parameter solid**
- **Hysteretic Damper For Lead-Rubber Bearing**
- **Hysteretic Damper For Friction Pendulum System**
- **User-Supplied Nonlinear Joint Properties**

Standard Linear Joint Model

Joint type: Spring stiffness only

The JOINT PROPERTIES data section defines linear spring stiffnesses for joints.

```
JOINT PROPERTIES  [N]
imat  < K_i >_i=1,N
```

imat	The material property identification number
K_i	**Elastic spring stiffness** corresponding to each local freedom. These local directions are defined for each joint element in the *LUSAS Element Reference Manual*.
N	Number of springs for joint element.

Example. Joint Properties

```
JPH3 ELEMENT TOPOLOGY
2 2 3 4
JOINT PROPERTIES
1 51.2 48.9 23.1
MATERIAL ASSIGNMENT
2 0 0 1
```

General Linear Joint Model

Joint type: General properties

The JOINT PROPERTIES GENERAL data section defines the full joint properties of spring stiffness, mass, coefficient of linear expansion and damping factor.

```
JOINT PROPERTIES GENERAL  [N]
imat  < K_i M_i C_i α_i a_ri b_ri >_i=1,N  [mcode]
```

imat	Material property identification number
K_i	**Elastic spring stiffness** corresponding to (i)th local freedom.
M_i	**Mass** corresponding to (i)th local freedom.
C_i	**Viscocity coefficient** - damping corresponding to the (i)th local freedom
α_i	**Coefficient of thermal expansion** corresponding to (i)th local freedom.
a_{ri}	**Mass Rayleigh damping constant** corresponding to (i)th local freedom
b_{ri}	**Stiffness Rayleigh damping** constant corresponding to (i)th local freedom
mcode	An integer number which determines the position of a mass or masses. = 0 for mass between nodes (default) = 1 for mass at 1st node. = 2 for mass at 2nd node.

N The number of springs for joint element.

Notes

1. In order to input the damping values Ci OPTION 324 must be specified.
2. Modeller will automatically set OPTION 324 .
3. If all the Ci input values are specified as zero the Rayleigh damping parameters will be used to form the element damping matrix.

Standard Nonlinear Elasto-Plastic Joint Model

Joint type: Elasto-plastic (tension and compression equal)

The JOINT PROPERTIES NONLINEAR 31 data section defines the material properties for the standard elasto-plastic joint model. The model incorporates elasto-plasticity with isotropic hardening. Equal tension and compression yield conditions are assumed.

```
JOINT PROPERTIES NONLINEAR 31  [N]

imat  < Ki Mi αi ari bri Fyldi κi >i=1,N  [mcode]
```

imat Material property identification number

K_i **Elastic spring stiffness** corresponding to (i)th local freedom.

M_i **Mass** corresponding to (i)th local freedom.

α_i **Coefficient of thermal expansion** corresponding to (i)th local freedom.

a_{ri} **Mass Rayleigh damping constant** corresponding to (i)th local freedom

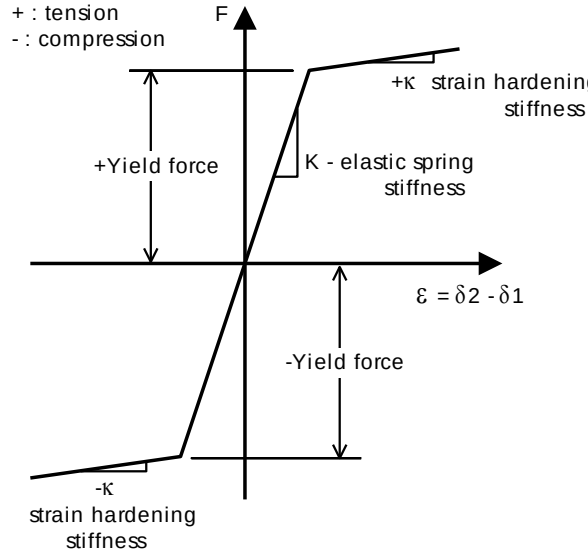
b_{ri} **Stiffness Rayleigh damping constant** corresponding to (i)th local freedom

$Fyld_i$ **Yield force**

κ_i **Hardening stiffness**

mcode An integer number which determines the position of a mass or masses.
= 0 for mass between nodes (default)
= 1 for mass at 1st node.
= 2 for mass at 2nd node.

N The number of springs for joint element.



Standard Nonlinear Elasto-Plastic Joint Model (Model 31)

General Nonlinear Elasto-Plastic Joint Model

Joint type: Elasto-plastic (tension and compression unequal)

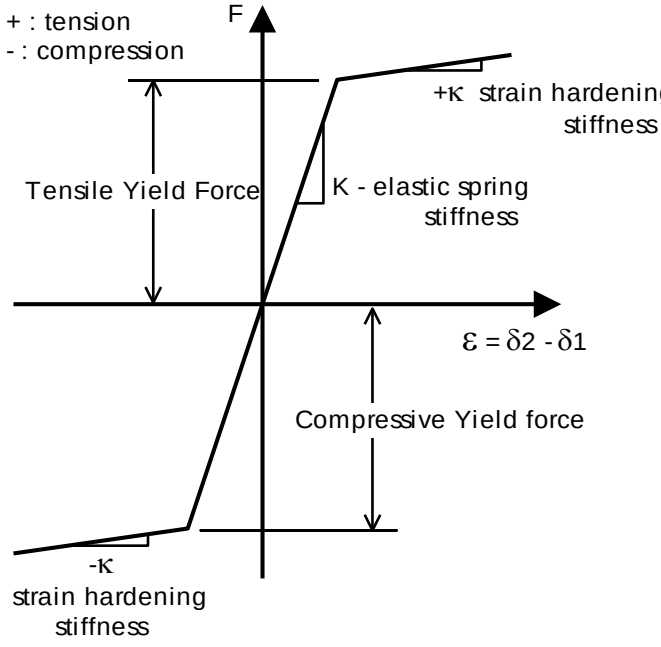
The JOINT PROPERTIES NONLINEAR 32 data section defines the material properties for the general elasto-plastic joint model. The model incorporates elasto-plasticity with isotropic hardening. Unequal tension and compression yield conditions are assumed.

```
JOINT PROPERTIES NONLINEAR 32  [N]

imat  < Ki Mi αi ari bri +Fyldi +κi -Fyldi -κi >i=1,N  [mcode]
```

imat	Material property identification number
K_i	**Elastic spring stiffness** corresponding to the (i)th local freedom.
M_i	**Mass** corresponding to (i)th local freedom.
α_i	**Coefficient of thermal expansion** corresponding to the (i)th local freedom.
a_{ri}	**Mass Rayleigh damping constant** corresponding to (i)th local freedom
b_{ri}	**Stiffness Rayleigh damping constant** corresponding to (i)th local freedom
$+Fyld_i$	**Tensile yield force**
$+\kappa_i$	**Tensile strain hardening stiffness**
$-Fyld_i$	**Compressive yield force** (value is always positive)
$-\kappa_i$	**Compressive strain hardening stiffness** (value is always positive)

mcode An integer number which determines the position of a mass or masses.
= 0 for mass between nodes (default)
= 1 for mass at 1st node.
= 2 for mass at 2nd node

N The number of springs for joint element.



General Nonlinear Elasto-Plastic Joint Model (Model 32)

Nonlinear Joint Model For Smooth Contact

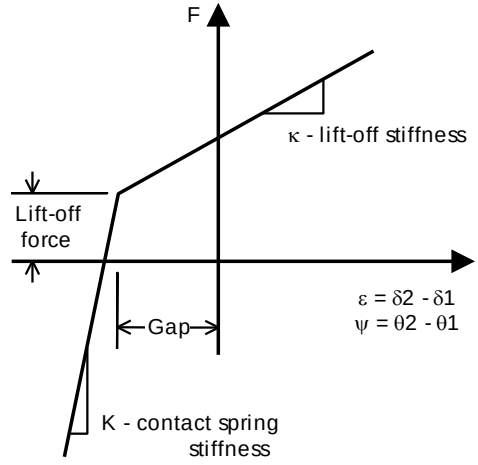
The JOINT PROPERTIES NONLINEAR 33 data section defines the material properties for the nonlinear joint model for smooth contact with an initial gap.

JOINT PROPERTIES NONLINEAR 33 [N]

imat < K_{ci} M_i α_i a_{ri} b_{ri} F_{yldi} κ_i gap_i >$_{i=1,N}$ [mcode]

imat The material property identification number
Kci The contact spring stiffness corresponding to (i)th local freedom.
Mi The mass corresponding to (i)th local freedom.
αi The coefficient of linear expansion corresponding to (i)th local freedom.
ari Mass Rayleigh damping constant corresponding to (i)th local freedom
bri Stiffness Rayleigh damping constant corresponding to (i)th local freedom
Fyldi Lift-off force

κi Lift-off stiffness
gapi The initial gap (see *Notes*)
mcode An integer number which determines the position of a mass or masses.
= 0 for mass between nodes (default)
= 1 for mass at 1st node.
= 2 for mass at 2nd node
N The number of springs for joint element.



Nonlinear Joint Model for Smooth Contact (Model 33)

Note

1. If an initial gap is used in a spring, then the positive local axis for this spring must go from node 1 to 2. If nodes 1 and 2 are coincident the relative displacement of the nodes in a local direction (d2- d1) must be negative to close an initial gap in that direction.

Nonlinear Joint Model For Frictional Contact

Joint type: Friction contact

The JOINT PROPERTIES NONLINEAR 34 data section defines the material properties for the nonlinear joint model for frictional contact with an initial gap.

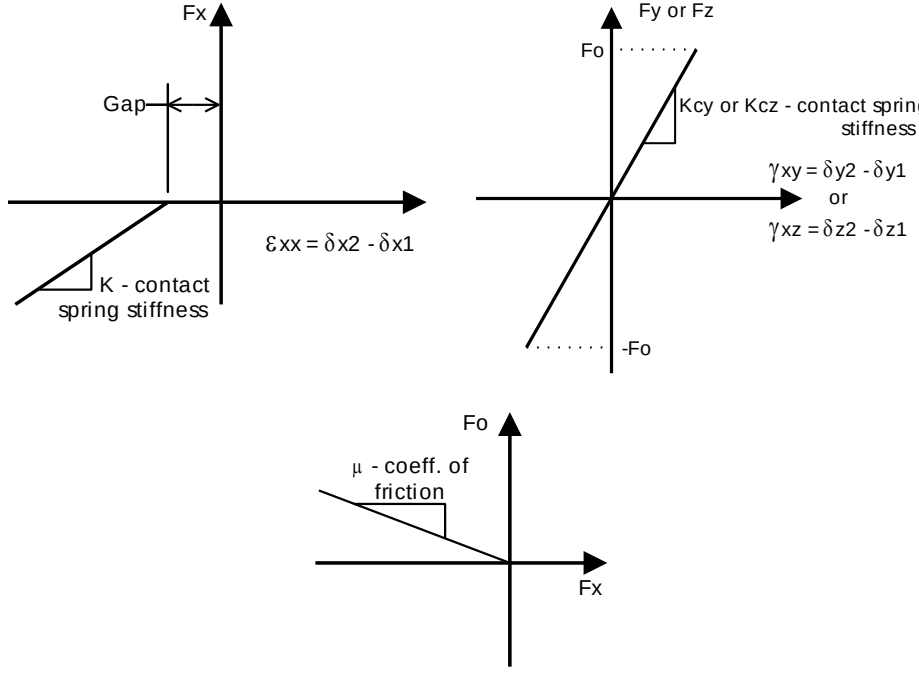
```
JOINT PROPERTIES NONLINEAR 34  [N]

imat  < Kci Mi αi ari bri >i=1,N  μ gap  [mcode]
```

imat Material property identification number
K_{ci} **Contact spring stiffness** corresponding to (i)th local freedom.
M_i **Mass** corresponding to (i)th local freedom.

α_i	**Coefficient of thermal expansion** corresponding to (i)th local freedom.
a_{ri}	**Mass Rayleigh damping constant** corresponding to (i)th local freedom
b_{ri}	**Stiffness Rayleigh damping constant** corresponding to (i)th local freedom
μ	**Coefficient of friction.**
gap	**Initial gap in x** The initial gap in the local x-direction.
mcode	An integer number which determines the position of a mass or masses. = 0 for mass between nodes (default) = 1 for mass at 1st node. = 2 for mass at 2nd node
N	The number of springs for joint element.

Notes

1. If an initial gap is used in a spring, then the positive local x axis for this spring must go from node 1 to 2. If nodes 1 and 2 are coincident the relative displacement of the nodes in the local x direction (dx2- dx1) must be negative to close an initial gap.



Nonlinear Joint Model for Frictional Contact (Model 34)

Visco-Elastic Dampers

Joint type: Viscous damper – Kelvin

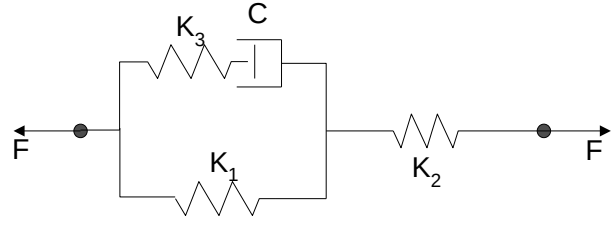
Joint type: Viscous damper - Four parameter solid

The JOINT PROPERTIES NONLINEAR 35 data section defines the material properties for a general preloaded visco-elastic damper joint model. The model consists of three springs and a dashpot in the so-called four-parameter solid model arrangement. Selectively setting the appropriate stiffnesses to zero allows the model to degenerate to a Kelvin (**K2**=0, **K3**=0) or Maxwell (**K1**=0, **K2** or **K3**=0) unit.

JOINT PROPERTIES NONLINEAR 35 [N]

imat < $K1_i$ M_i C_i α_i α_{vi} Fp_i $K2_i$ $K3_i$ >$_{i=1,N}$ [mcode]

imat	Material property identification number
$K1_i$	**Spring stiffness** corresponding to (i)th local freedom (see below).
M_i	**Mass** corresponding to (i)th local freedom.
C_i	**Viscosity coefficient** Damping corresponding to the (i)th local freedom
α_i	**Coefficient of thermal expansion** corresponding to (i)th local freedom.
α_{vi}	**Velocity exponent** corresponding to (i)th local freedom
Fp_i	**Preload force** corresponding to (i)th local freedom
$K2_i$	**Spring stiffness** corresponding to (i)th local freedom in parallel with damper (see below).
$K3_i$	**Spring stiffness** corresponding to (i)th local freedom in series with damper (see below).
mcode	An integer number which determines the position of a mass or masses. = 0 for mass between nodes (default) = 1 for mass at 1st node. = 2 for mass at 2nd node.
N	The number of springs for joint element.



Four parameter solid model for visco-elasticity (Model 35)

Notes

1. Stiffnesses, mass and preload force can be set to zero in order to create a viscous damper only.
2. The preload force represents the force that must be exceeded to cause movement of the damper piston (typically the initial pressure in hydraulic fluid).
3. If a Kelvin unit is defined the damping force is computed from $F = Cv^{\alpha v}$.
4. The viscosity coefficient is typically evaluated from $C = Fn/Vn^{\alpha v}$ where *Fn* and *Vn* are the maximum force and velocity that are expected to occur in the damper.

Lead-Rubber Bearing with Hysteretic damping

Joint type: Lead rubber bearing

Lead-rubber bearings (LRBs) are generally constructed of low-damping natural rubber with a preformed central hole, into which a lead core is press-fitted. The central lead core provides an additional means of energy dissipation and the energy absorbed by the core reduces the lateral displacement of the isolator. This system provides the combined features of vertical load support, horizontal flexibility, restoring force and damping in a single unit.

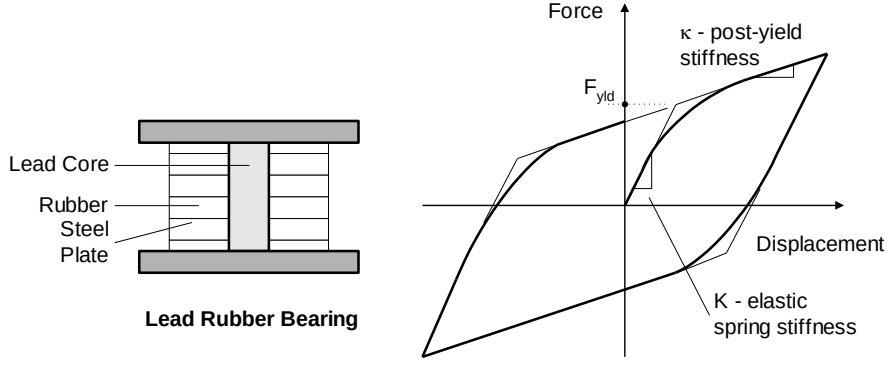
Under lateral deformation, the lead core deforms in almost pure shear, yields at low level of stress (approximately 8 to 10 MPa in shear at normal temperature), and produces hysteretic behavior that is stable over many cycles. Unlike mild steel, lead recrystallizes at normal temperature (about 20°C), so that repeated yielding does not cause fatigue failure. LRBs generally exhibit characteristic strength that ensures rigidity under service loads.

JOINT PROPERTIES NONLINEAR 36 [N]

imat <K_i M_i C_i α_i a_{ri} b_{ri} F_{yld_i} κ_i A_i s_i>$_{i=1,N}$ λ_{cp} β γ [mcode]

imat	Material property identification number
K_i	**Elastic spring stiffness** corresponding to the *i*th local freedom
M_i	**Mass** corresponding to the *i*th local freedom
C_i	**Viscosity coefficient** Damping corresponding to the (i)th local freedom
α_i	**Coefficient of thermal expansion** corresponding to the *i*th local freedom
a_{ri}	**Mass Rayleigh damping constant** corresponding to the *i*th local freedom
b_{ri}	**Stiffness Rayleigh damping constant** corresponding to the *i*th local freedom
F_{yld_i}	**Yield force** (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D)
κ_i	**Post-yield stiffness** (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D)

$\mathbf{A}_i$	**Hysteretic parameter A** (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D)
$\mathbf{s}_i$	**Hysteretic parameter s** (see Notes)
λ_{cp}	**Biaxial coupling coefficient** used in evolution of hysteretic variables (only applicable for the 3D case – see Notes)
β	**Hysteretic parameter beta**
γ	**Hysteretic parameter gamma**
mcode	An integer number which determines the position of a mass or masses. =0 for mass between nodes (default) =1 for mass at 1st node =2 for mass at 2nd node
N	Number of springs for joint element



Schematic LRB and hysteretic behaviour (Model 36)

Notes

1. The input parameters are utilised in the following manner; for the 3D case, the forces in the damper are computed from:

$$F_x = \kappa_x u + \left(1-\kappa_x/K_x\right)F_{yld_x} z_x$$

$$F_y = \kappa_y v + \left(1-\kappa_y/K_y\right)F_{yld_y} z_y$$

where the evolution of the hysteretic terms is given by:

$$\begin{bmatrix} u^y \dot{z}_x \\ v^y \dot{z}_y \end{bmatrix} = \begin{bmatrix} A_x - z_x^{s_x}\left\{\gamma \mathrm{sgn}\left(\dot{u}z_x\right)+\beta\right\} & -\lambda_{cp} z_x z_y \left\{\gamma \mathrm{sgn}\left(\dot{v}z_y\right)+\beta\right\} \\ -\lambda_{cp} z_x z_y \left\{\gamma \mathrm{sgn}\left(\dot{u}z_x\right)+\beta\right\} & A_y - z_y^{s_y}\left\{\gamma \mathrm{sgn}\left(\dot{v}z_y\right)+\beta\right\} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}$$

where, u^y and v^y are displacements when yield occurs. For the 3D uncoupled case (λ_{cp}=0), the values for $\mathbf{s_x}$ and $\mathbf{s_y}$ are taken as the specified input parameters. For the coupled case (λ_{cp}>0) the values are fixed, $\mathbf{s_x}$=$\mathbf{s_y}$=2. z_x and z_y are dimensionless hysteretic variables bounded by values ± 1.

When working in 2D, the input parameter $\mathbf{s_x}$ should take a value > 1, where it is used to define the following hysteretic term:

$$z_X = s_X\sqrt{\frac{1}{\gamma+\beta}}$$

Note that the value for the hysteretic coupling parameter, λ_{cp}, must lie within the limits of 0 and 1; a value of 0 leads to fully uncoupled hysteretic equations and a value of 1 fully coupled equations. For more information please consult the *LUSAS Theory Manual.*

2. The input parameters **F_{yld}, κ, A, s** and the other hysteretic control parameters are only applicable to the joint local x translation in 2D and the local x and y translation in 3D.
3. Lift-off does not occur in this model and the vertical stiffness is taken as the local y direction in 2D and the local z direction in 3D.
4. If all the C_i input values are specified as zero the Rayleigh damping parameters will be used to form the element damping matrix.

Sliding/Frictional Pendulum System with Hysteretic Damping

Joint type: Friction pendulum system

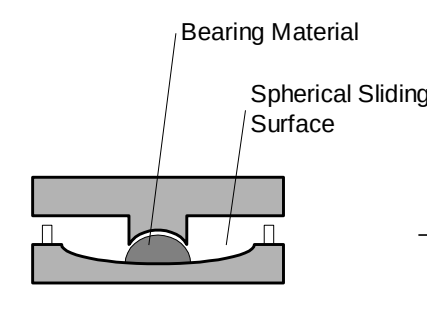
A friction pendulum system (FPS) bearing consists of a spherical sliding surface and an articulated slider which is faced with a high pressure capacity bearing material. The bearing may be installed as shown below or upside down with the spherical surface facing downwards. Irrespective of the installation method, the behaviour is identical.

```
JOINT PROPERTIES NONLINEAR 37 [N]
```

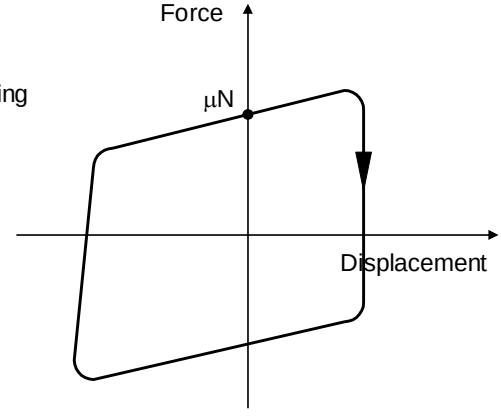
imat $<K_i\ M_i\ C_i\ \alpha_i\ a_{ri}\ b_{ri}\ \mu_{max0_i}\ \mu_{maxp_i}\ \mu_{min_i}\ \varphi_i\ \varepsilon_i\ A_i\ s_i>_{i=1,N}\ A_c\ R\ \lambda_{cp}\ \beta\ \gamma$ **[mcode]**

imat	Material property identification number
K_i	**Elastic spring stiffness** corresponding to the *i*th local freedom
M_i	**Mass** corresponding to the *i*th local freedom
C_i	**Viscosity coefficient** Damping corresponding to the (i)th local freedom
α_i	**Coefficient of thermal expansion** corresponding to the *i*th local freedom
a_{ri}	**Mass Rayleigh damping constant** corresponding to the *i*th local freedom
b_{ri}	**Stiffness Rayleigh damping constant** corresponding to the *i*th local freedom
μ_{max0_i}	**Friction coefficient at zero pressure** (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D).
μ_{maxp_i}	**Friction coefficient at high pressure** (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D).

μ_{min_i}	**Friction coefficient at (near) zero velocity** (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D).
φ_i	**Coefficient of velocity** Parameter controlling the variation of the coefficient of friction with velocity (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D).
ε_i	**Pressure coefficient** Parameter used to determine the variation of the friction coefficient with pressure (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D)
A_i	**Hysteretic parameter A** (only applicable to the 1st local freedom in 2D and the 1st and 2nd local freedoms in 3D)
s_i	**Hysteretic parameter s** (see Notes)
A_c	**Contact area** of FPS
R	**Radius of sliding surface** (spherical) Specify R=0 for a flat surface.
λ_{cp}	**Biaxial coupling coefficient** Parameter used in evolution of hysteretic variables (only applicable for the 3D case – see Notes)
β	**Hysteretic parameter beta**
γ	**Hysteretic parameter gamma**
mcode	An integer number which determines the position of a mass or masses. =0 for mass between nodes (default) =1 for mass at 1st node =2 for mass at 2nd node
N	Number of springs for joint element



Friction Pendulum Bearing



FPS layout and schematic representation (Model 37)

Notes

1. The input parameters are utilised in the following manner; for the 3D case, the forces in the FPS element, including the restorative forces, are given by:

$$F_x = \frac{N}{R}u + \mu_s N z_x$$

$$F_y = \frac{N}{R}v + \mu_s N z_y$$

where the normal force N is evaluated as follows

$N=K_z w$

where K_z and w are stiffness and compressive deformation in the contact direction. The friction coefficient μ_s is given by:

$$\mu_s = \mu_{max} - (\mu_{max} - \mu_{min})\exp(-\varphi|\dot{u}|)$$

where μ_{max} is computed from:

$$\mu_{max} = \mu_{max0} - (\mu_{max0} - \mu_{maxp})\tanh(\varepsilon P)$$

and the bearing pressure is given by:

$P=N/A_c$

For the 3D case the evolution of the hysteretic terms is given by:

$$\begin{bmatrix} u^y \dot{z}_x \\ v^y \dot{z}_y \end{bmatrix} = \begin{bmatrix} A_x - z_x^{s_x}\{\gamma \mathrm{sgn}(\dot{u} z_x) + \beta\} & -\lambda_{cp} z_x z_y \{\gamma \mathrm{sgn}(\dot{v} z_y) + \beta\} \\ -\lambda_{cp} z_x z_y \{\gamma \mathrm{sgn}(\dot{u} z_x) + \beta\} & A_y - z_y^{s_y}\{\gamma \mathrm{sgn}(\dot{v} z_y) + \beta\} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}$$

where, u^y and v^y are displacements when yield occurs. For the 3D uncoupled case ($\lambda_{cp}=0$), the values for $\mathbf{s_x}$ and $\mathbf{s_y}$ are taken as the specified input parameters. For the coupled case ($\lambda_{cp}>0$) the values are fixed, $\mathbf{s_x}=\mathbf{s_y}=2$. z_x and z_y are dimensionless hysteretic variables bounded by values ± 1.

When working in 2D, the input parameter $\mathbf{s_x}$ should take a value > 1, where it is used to define the following hysteretic term:

$$z_x = \sqrt[s_x]{\frac{1}{\gamma + \beta}}$$

Note that the value for the hysteretic coupling parameter, λ_{cp}, must lie within the limits of 0 and 1; a value of 0 leads to fully uncoupled hysteretic equations and a value of 1 fully coupled equations. For more information please consult the *LUSAS Theory Manual*.

2. The input parameters μ_{max0}, μ_{maxp}, μ_{min}, φ, ε, **A**, **s** and the other hysteretic control parameters are only applicable to the joint local x translation in 2D and the local x and y translation in 3D.

3. Lift-off occurs when the local vertical strain (ε_y in 2D, ε_z in 3D) is greater than zero. If lift-off occurs the vertical stiffness is set to zero and the hysteretic terms are also initialised in readiness for re-contact. In a compressive state the vertical stiffness is taken as the input value K_y in 2D and K_z in 3D.

4. If all the $\mathbf{C_i}$ input values are specified as zero the Rayleigh damping parameters will be used to form the element damping matrix.

User-Supplied Nonlinear Joint Properties

Joint type: Nonlinear user

The USER joint model facility allows user-supplied joint property subroutines to be used from within LUSAS. This facility provides controlled access to the pre- and post-solution constitutive processing and nonlinear state variable output via these user-supplied subroutines.

By default these routines are supplied as empty routines with defined interfaces that are unchangeable. The externally developed code should be placed into these routines which is then linked into the LUSAS system. Source code access is available to these interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. See *Appendix C: Solver User Interface Routines.*

Since user specification of a constitutive model involves the external development of source FORTRAN code, as well as access to LUSAS code, this facility is aimed at the advanced LUSAS user.

The required joint properties input for the user-supplied joint model is completely general. However, the properties $\mathbf{M_i, C_i, \alpha_i, a_{ri}, b_{ri}}$ along with **mcode** are reserved for LUSAS internal use: you only need to utilise them if other features of the program are required (e.g. dynamic and thermal analyses). The data section JOINT PROPERTIES NONLINEAR USER is used to define all the joint property parameters for user-supplied joint models.

JOINT PROPERTIES NONLINEAR USER LPTUSR N NPRZS NPRZJ NSTAT

imat $<K_i\ M_i\ C_i\ \alpha_i\ a_{ri}\ b_{ri}\ \{u_j\}_{j=1,nprzs-6}>_{i=1,N}\ \{p_k\}_{k=1,nprzj}$ **[mcode]**

LPTUSR	**User model number**
N	Number of springs for joint element
NPRZS	Total number of joint properties per spring
NPRZJ	Number of properties per joint (common to all springs)
NSTAT	**Number of state variables**
imat	Material property identification number
K_i	**Elastic spring stiffness** corresponding to the *i*th local freedom
M_i	**Mass** corresponding to the *i*th local freedom
C_i	**Viscocity coefficient** Damping coefficient corresponding to the *i*th local freedom
α_i	**Coefficient of thermal expansion** corresponding to the *i*th local freedom
a_{ri}	**Mass Rayleigh damping constant** corresponding to the *i*th local freedom
b_{ri}	**Stiffness Rayleigh damping constant** corresponding to the *i*th local freedom

u_j	User defined material properties for individual component directions
P_k	User defined joint properties common to all directions
mcode	An integer number which determines the position of a mass or masses. =0 for mass between nodes (default) =1 for mass at 1st node =2 for mass at 2nd node

Notes

1. The total number of joint properties, NPRZ, will be computed from N*NPRZS + NPRZJ + 1.
2. The number of material properties input must be equal to that specified in the data.
3. When no state variables are required (i.e. when nstat is specified as zero) a warning message will be invoked.
4. The properties $M_i, C_i, \alpha_i, a_{ri}, b_{ri}$ along with **mcode** are required by LUSAS in order to perform other types of analyses: for example thermal problems require the coefficients of thermal expansion α_i, and a dynamic analysis requires the mass M_i and possibly damping via C_i or a_{ri} and b_{ri} along with **mcode**.
5. If all the C_i input values are specified as zero the Rayleigh damping parameters will be used to form the element damping matrix.
6. Option 179 can be set for argument verification within the user routines.

Material Properties Mass

The data section MATERIAL PROPERTIES MASS is used to specify the element nodal masses in the local coordinate system for non-structural mass elements.

MATERIAL PROPERTIES MASS naxes nnode

imat mx_i my_i {mz_i} ... mx_{nnod} my_{nnod} {mz_{nnode}}

imat	The material property identification number
naxes	Number of axes in structure (2D or 3D)
nnode	Number of nodes to element
mx_i	Mass at node i in local x direction
my_i	Mass at node i in local y direction
mz_i	Mass at node i in local z direction (only if naxes=3)

Notes

1. MATERIAL PROPERTIES MASS must only be assigned to non-structural mass elements. Only translational masses in the x, y and z directions are available. Rotational masses are not available.
2. For point elements the nodal (point) mass is input. For line elements the mass per unit length is input. For surface elements the mass per unit area is input.
3. By default nodal masses are defined with respect to local element axes. However, it is also possible to define the mass orientation with respect to any CARTESIAN SET. This is specified under MATERIAL ASSIGNMENTS.

Nonlinear Material Properties

The nonlinear models may be used to model yielding of materials such as metals, concrete, rubber, soils and rocks etc.. The following types of nonlinear material models are available (described in detail in the *LUSAS Theory Manual*):

- **Plasticity Models**
  - Plastic Definition (von Mises, Hill, Hoffman)
  - Optimised Von Mises Model
  - Drucker-Prager Model
  - Tresca Model
  - Non-Associative Mohr-Coulomb Model
  - Multi Crack Concrete Model
  - Elasto-Plastic Stress Resultant Model
  - User-Supplied Nonlinear Material Properties
- **Viscous Definition (Creep, Viscoelasticity)**
- **Damage Model**
- **Composite Matrix Failure Model**
- **Two Phase Material Model**
- **Rubber Model**
- **Volumetric Crushing Model**
- **Concrete Creep Models**
  - CEB-FIP Model Code 1990
  - Chinese Creep Code for Dams
- **Generic Polymer Model**
- **2D Interface Model / 3D Interface Model**
- **Delamination Model**
- **Resin Cure Model**
- **Shrinkage**

A full section (stress resultant) yield criterion is available for some of the shell and beam elements. Other elasto-plastic models for Gauss point stress evaluations are applicable to shell, membrane and solid elements. Various yield criteria with isotropic and kinematic hardening are available. Implicit integration of the elasto-plastic constitutive equations is also implemented for some of the models. The pressure dependent material models allow for different properties in tension and compression.

In addition to the standard element data output, the following details of the nonlinear material model behaviour are also output (at the Gauss points):

YFUNC The value of the yield function.
zero: stress on the yield surface.

negative: stress within the yield surface.

CURYS The value of the equivalent yield stress.

EPSTN The value of the equivalent plastic strain.

IYLD The value of the yield flag.
0 - elastic
1 - plastic

NSTEPS The number of sub increments used to integrate the elasto-plastic strain increment (explicit integration), or the number of Newton-Raphson iterations used to satisfy the elasto-plastic constitutive relations (implicit integration).

Plastic Definition

The PLASTIC DEFINITION data chapter allows more flexibility in the way that plastic properties for a nonlinear material can be defined. In particular, the following options may be used to define a hardening curve by specifying the following pairs of values:

- the gradient of the curve together with the plastic strain limit for which the gradient applies
- the yield stress together with a plastic strain value
- the yield stress together with a total strain value

A PLASTIC DEFINITION should be used in conjunction with a linear material property; the linear and nonlinear sets of properties are then assigned to elements using the MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL data chapters. The following linear material property types can be assigned to the same elements as a PLASTIC DEFINITION data set:

- MATERIAL PROPERTIES
- MATERIAL PROPERTIES ORTHOTROPIC
- MATERIAL PROPERTIES ORTHOTROPIC PLANE STRAIN
- MATERIAL PROPERTIES ORTHOTROPIC THICK
- MATERIAL PROPERTIES ORTHOTROPIC AXISYMMETRIC
- MATERIAL PROPERTIES ORTHOTROPIC SOLID

Note. If an orthotropic linear material is used to define the linear elastic properties, then any accompanying plastic data set must include an orthotropic stress potential.

STRESS POTENTIAL {VON_MISES | HILL | HOFFMAN}

ipls [σ_{xx} σ_{yy} σ_{zz} σ_{xy} σ_{yz} σ_{zx} T | σ_{xx}^{t} σ_{xx}^{c} σ_{yy}^{t} σ_{yy}^{c} σ_{zz}^{t} σ_{zz}^{c} σ_{xy} σ_{yz} σ_{xz} T]

YIELD STRESS [n_{σ}]

ipls σ_{yo}^{t} [σ_{yo}^{c}] [T]

HARDENING CURVE [HARDENING_GRADIENT | PLASTIC_STRAIN | TOTAL_STRAIN] N^t [N^c]

ipls < H_i^t S_i^t >$_{i=1,Nt}$ [< H_i^c S_i^c >$_{i=1,Nc}$] [T]

HEAT FRACTION

ipls h_f [T]

ipls	The plastic definition set identification number, see *Notes*.
σij	Stresses defining the yield surface for Hill model.
σijt(c)	Stresses defining the yield surface in tension (compression) for Hoffman model.
T	Reference temperature.
Nσ	Number of yield stresses to be specified, see *Notes*. (Default=1).
σy0t(c)	Initial reference tensile (compressive) yield stress.
Nt(c)	Number of points/sections defining the tensile (compressive) hardening curve.
Hit(c)	First value defining the hardening curve for tension (compression), see *Notes*.
Sit(c)	Second value defining the hardening curve for tension (compression), see *Notes*.
hf	Heat fraction.

Notes

1. Data sections specified under the PLASTIC DEFINITION data chapter, which are used to build up the definition of a particular material, must be allocated the same plastic definition identification number, **ipls**.
2. The STRESS POTENTIAL data section must always be used. A stress potential must be specified for every set of plastic material properties defined.
3. The data sections under the PLASTIC DEFINITION chapter may be specified in any order.
4. When using the HARDENING CURVE, YIELD STRESS or HEAT FRACTION data sections a linear material model must be specified to define the linear material properties.
5. The PLASTIC DEFINITION data must be combined with other material data to define the required nonlinear material using the MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL data chapters.
6. The PLASTIC DEFINITION data chapter is intended to be used in conjunction with linear material models (e.g. MATERIAL PROPERTIES ORTHOTROPIC).
7. If tables of properties in the separate data sections defining a particular material are inconsistent, an amalgamated table is assembled for interpolation. Tables are inconsistent if they are made up of a different number of lines of data or use different reference temperatures. If this is the case, a value may need to be interpolated which is outside the temperature bounds of the table defined in a particular data section. If this occurs the appropriate extreme value in the table is

used. If Option 227 is invoked by the user, this occurrence will cause a data error and the analysis will be terminated.

8. For additional information see:

   Stress Potential
   Yield Stress
   Hardening Curve
   Heat Fraction

Stress Potential

The use of nonlinear material properties applicable to a general multi-axial stress state requires the specification of yield stresses in each direction of the stress space when defining the yield surface (see the *LUSAS Theory Manual*). These stresses are specified under the STRESS POTENTIAL command and are assigned to appropriate elements through the MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL data chapters. Hill, von Mises and Hoffman nonlinear material models are available.

Notes

1. The STRESS POTENTIAL data section must always be used. A stress potential must be specified for every set of plastic material properties defined.
2. If a stress potential type is not specified then von Mises is set as default.
3. The yield surface must be defined in full, irrespective of the type of analysis undertaken. This means that none of the stresses defining the yield surface can be set to zero. For example, in a plane stress analysis, the out of plane direct stress **σzz**, must be given a value which physically represents the model to be analysed.
4. The von Mises stress potential does not require the input of any parameters.
5. The stresses defining the yield surface in both tension, **σijt**, and compression, **σijc**, for the Hoffman potential must be positive.
6. An orthotropic material property must be assigned with the Hill or Hoffman stress potentials.
7. The *LUSAS Theory Manual* should be consulted if further information on these stress potential parameters is required.
8. The STRESS POTENTIAL data section can be specified under the PLASTIC DEFINITION or VISCOUS DEFINITION data chapters. If specified under the VISCOUS DEFINITION data chapter, the material properties must be linear and the STRESS POTENTIAL parameters are applied to the creep properties only. If creep is defined together with a nonlinear material property, the STRESS POTENTIAL parameters will be applied to both the creep and plasticity; in this instance, the parameters must only be specified under the PLASTIC DEFINITION data chapter.

Yield Stress

The stress or stresses specified under the YIELD STRESS data section define an initial uniaxial yield stress. For orthotropic material models, this value is only required for the definition of isotropic hardening; the current stress state for such models is computed using the yield surface defined under the STRESS POTENTIAL data section.

Notes

1. The number of yield stresses to be specified depends on the material model defined. If different yield stresses are required in tension and compression, then **nσ** must be set to 2. (Default **nσ**=1).
2. There are some occasions when there is no need to use the YIELD STRESS data section:
3. When using the PLASTIC_STRAIN or TOTAL_STRAIN options to define the hardening curve. (The first stress in the curve data is taken as the uniaxial yield stress).
4. When using an orthotropic material property without specifying a hardening curve. In this instance LUSAS sets an arbitrary value of 1 and writes it to the output file.
5. When using the YIELD STRESS data section a linear material model must be used to specify the linear material properties.
6. The YIELD STRESS data section must always be used if the HARDENING_GRADIENT option is used to define the hardening curve.

Hardening Curve

The isotropic hardening behaviour for a nonlinear material can be defined under the HARDENING CURVE data section. There are three options available for defining the curve by using the HARDENING_GRADIENT, PLASTIC_STRAIN or TOTAL_STRAIN data sections. The data input required for each option is described below.

Notes

1. The values used to define the hardening curve depend upon the option chosen for input.
   - HARDENING_GRADIENT:
     - **Hit(c)** the slope of the (i)th section of the reference tensile (compressive) yield stress against effective plastic strain curve.
     - **Sit(c)** the limit on the effective plastic strain up to which the (i)th section of the hardening curve for tension (compression) is valid.
   - PLASTIC_STRAIN:

Hit(c) the (i)th tensile (compressive) uniaxial yield stress defining the hardening curve.

Sit(c) the effective plastic strain corresponding with the (i)th tensile (compressive) uniaxial yield stress.

- TOTAL_STRAIN:

Hit(c) the (i)th tensile (compressive) uniaxial yield stress defining the hardening curve.

Sit(c) the total strain corresponding with the (i)th tensile (compressive) uniaxial yield stress.

2. When using the PLASTIC_STRAIN or TOTAL_STRAIN options the first pair of values defining the curve must relate to the initial yield point. In this instance the YIELD STRESS data section can be omitted.
3. All values defined under PLASTIC_STRAIN or TOTAL_STRAIN data sections must be positive, even when defining values for compression.
4. When using the HARDENING CURVE data section a linear material model must be used to specify the linear material properties.
5. The YIELD STRESS data section must always be used if the HARDENING_GRADIENT option is used to define the hardening curve.

Heat Fraction

The heat fraction is only applicable in thermo-mechanical coupled analyses where the heat produced due to the rate of generation of plastic work is of interest. The HEAT FRACTION data section is used to define the fraction of plastic work that is converted to heat.

Notes

1. When using the HEAT FRACTION data section a linear material model must be used to specify the linear material properties.
2. The heat fraction should take a value between 0.0 and 1.0.

Example of Plastic Definition

Example. Plastic Definition

```
MATERIAL PROPERTIES
1  2E9 0.25  7E3  2E-5  5.0  1E-3
PLASTIC DEFINITION
STRESS POTENTIAL VON_MISES
23
```

```
HARDENING CURVE HARD 4 4
23...
200.0 0.005 190.0 0.01 150.0 0.015 100.0 0.02...
250.0 0.005 230.0 0.01 200.0 0.015 160.0 0.02
YIELD STRESS 2
23...
2E8   1.9E8
HEAT FRACTION
23...
0.9
MATERIAL ASSIGNMENTS
1 10 1   1 0 23
```

Optimised Von Mises Model

The elasto-plastic von Mises yield surface models may be used to represent ductile behaviour of materials which exhibit little volumetric strain (for example, metals).

In the optimised implicit (backward Euler) model the direction of plastic flow is evaluated from the stress return path. The implicit method allows the proper definition of a tangent stiffness matrix which maintains the quadratic convergence of the Newton-Raphson iteration scheme otherwise lost with the explicit method. This allows larger load steps to be taken with faster convergence. For most applications, the implicit method should be preferred to the explicit method.

The data section MATERIAL PROPERTIES NONLINEAR 75 is used to define the material properties for the optimised implicit backward Euler von Mises material model. The model incorporates linear isotropic and kinematic hardening.

MATERIAL PROPERTIES NONLINEAR 75 N

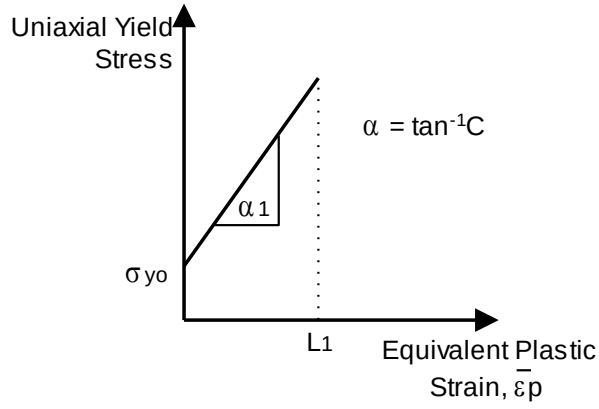
imat E ν ρ α a_r b_r h_f T σ_{yo} H_{I1} H_{K1} L_1

imat	The material property identification number
E	Young's modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
hf	Heat fraction coefficient (see *Notes*)
T	Reference temperature

σyo	Initial uniaxial yield stress
HI1	Isotropic hardening parameter (see *Notes*)
HK1	Kinematic hardening parameter (see *Notes*)
L1	The limit on the equivalent plastic strain up to which the hardening parameters are valid
N	The number of straight line approximations to the hardening curve (N must equal 1 for this model)

Notes

1. See Nonlinear Material Hardening Convention for an example of how to convert from the elasto-plastic modulus, **Ep**, to the slope of the uniaxial yield stress against equivalent plastic strain curve when specifying a hardening curve.
2. The heat fraction coefficient represents the fraction of plastic work which is converted to heat and takes a value between 0 and 1. For compatibility with pre LUSAS 12 data files specify Option -235.



Nonlinear Hardening Curve for the Backward Euler von Mises and Hill Models (Model 75)

Drucker-Prager Model

The Drucker-Prager elasto-plastic model (see figures below) may be used to represent the ductile behaviour of materials which exhibit volumetric plastic strain (for example, granular materials such as concrete, rock and soils). The model incorporates isotropic hardening.

The data section MATERIAL PROPERTIES NONLINEAR 64 is used to define the material properties.

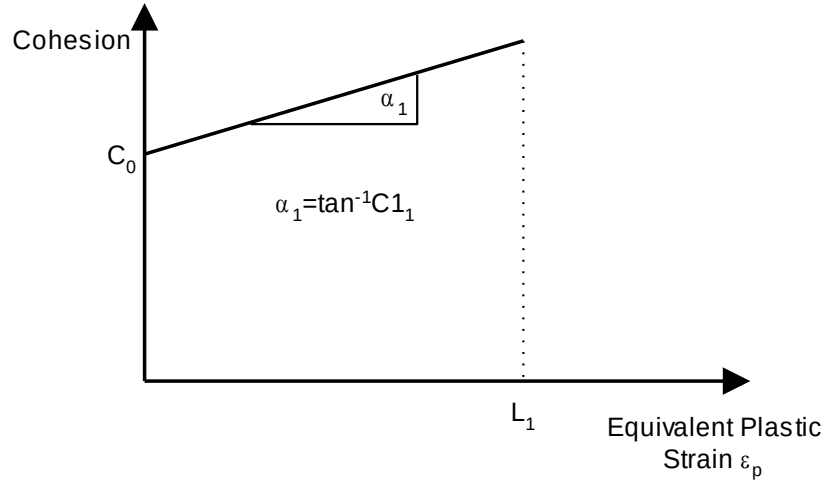
```
MATERIAL PROPERTIES NONLINEAR 64  N
```

imat **E** **ν** **ρ** **α** **a_r** **b_r** **h_f** **T** **c_0** **ϕ_0** **$C1_1$** **$C2_1$** **L_1**

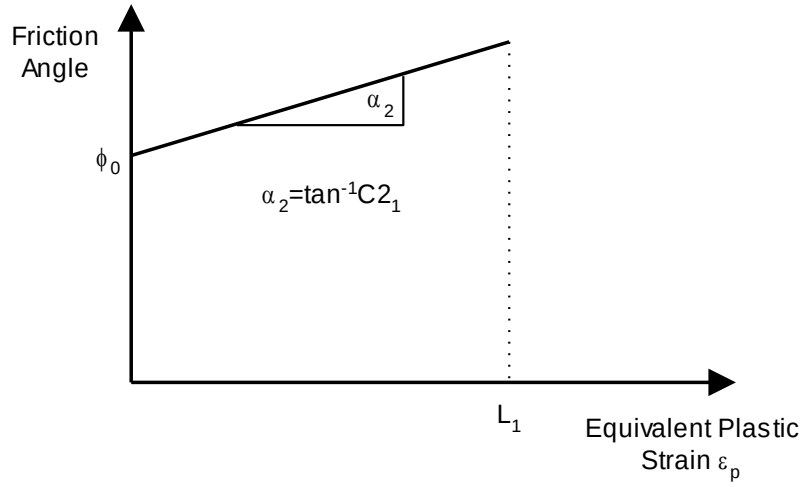
imat	The material property identification number
E	Young's modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
hf	Heat fraction coefficient (see *Notes*)
T	Reference temperature
c0	Initial cohesion (see *Notes*)
f0	Initial friction angle (degrees)
C11	The slope of the cohesion against the equivalent plastic strain
C21	The slope of the friction angle against the equivalent plastic strain
L1	The limit to which the hardening curve is valid
N	The number of straight line approximations to the hardening curve (**N** must equal 1)

Notes

1. The heat fraction coefficient represents the fraction of plastic work which is converted to heat and takes a value between 0 and 1. For compatibility with pre LUSAS 12 data files specify Option -235.
2. Setting the initial cohesion (**c0**) to zero is not recommended as this could cause numerical instability under certain loading conditions.



Cohesion Definition for the Drucker-Prager Yield Model (Model 64)



Friction Angle Definition for the Drucker-Prager Yield Model (Model 64)

Tresca Model

The elasto-plastic Tresca yield surface model may be used to represent ductile behaviour of materials which exhibit little volumetric strain (for example, metals). The model incorporates elasto-plastic behaviour with isotropic hardening. The data section MATERIAL PROPERTIES NONLINEAR 61 is used to define the material properties for the Tresca yield surface model.

MATERIAL PROPERTIES NONLINEAR 61 N

imat E ν ρ α a_r b_r h_f T σ_{yo} C_1 L_1

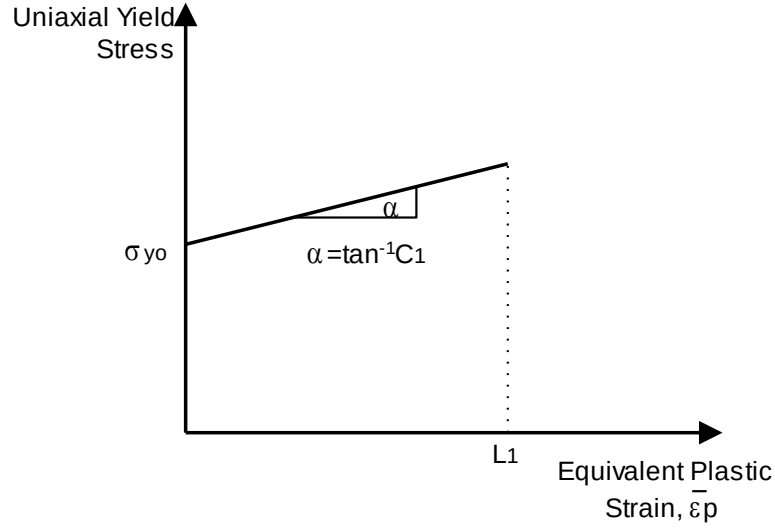
imat	The material property identification number
E	Young's modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
hf	Heat fraction coefficient (see *Notes*)
T	Reference temperature
σyo	Initial uniaxial yield stress
C1	The slope of the uniaxial yield stress against equivalent plastic strain (see *Notes*)

L1 The limit on the equivalent plastic strain up to which the hardening curve is valid

N The number of straight line approximations to the hardening curve (N must equal 1 for this model)

Notes

1. See Nonlinear Material Hardening Convention for an example of how to convert from the elasto-plastic modulus, Ep, to the slope of the uniaxial yield stress against equivalent plastic strain curve when specifying a hardening curve.
2. The heat fraction coefficient represents the fraction of plastic work which is converted to heat and takes a value between 0 and 1. For compatibility with pre LUSAS 12 data files specify Option -235.



Hardening Curve Definition for the Tresca and von Mises Yield Models (Models 61 & 62)

Non-Associated Mohr-Coulomb Model

The non-associated Mohr-Coulomb elasto-plastic model (see figures below) may be used to represent dilatant frictional materials which increasing shear strength with increasing confining stress (for example, materials such as granular soils and rock). The model incorporates isotropic hardening.

The data section MATERIAL PROPERTIES NONLINEAR 65 is used to define the material properties.

```
MATERIAL PROPERTIES NONLINEAR 65

imat  E ν ρ α ar br T ci ϕi ϕf ψ hc εf
```

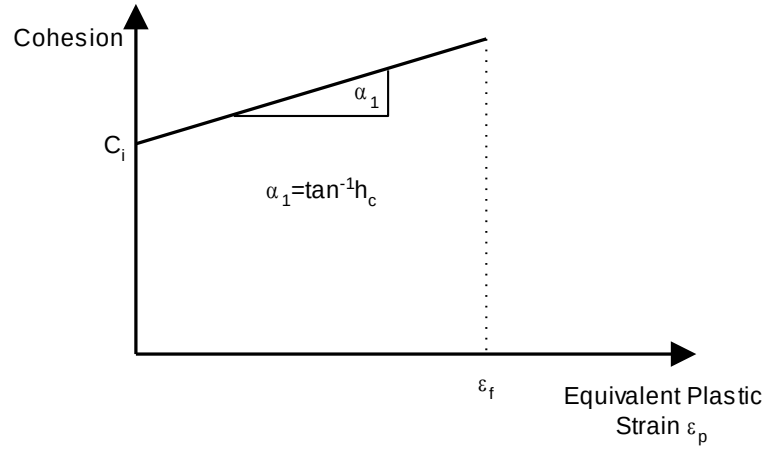
imat	The material property identification number
E	Young's modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
T	Reference temperature
ci	Initial cohesion
ϕi	Initial friction angle (degrees)
ϕf	Final friction angle (degrees)
ψ	Dilation angle (degrees)
hc	Cohesion hardening parameter
εf	Limiting equivalent plastic strain

Notes

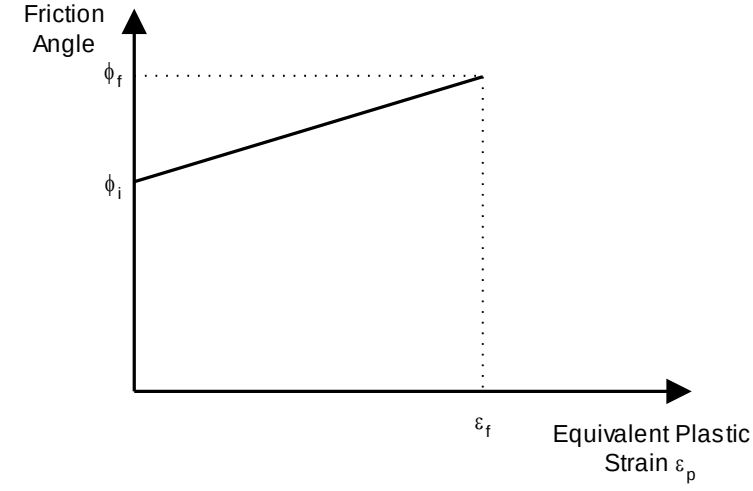
1. The non-associated Mohr-Coulomb model may be used with 2D and 3D continuum elements, 2D and 3D explicit dynamics elements, solid composite elements and semiloof or thick shells.
2. The non-symmetric solver (Option 64) is automatically switched on when using the non-associated Mohr-Coulomb model.
3. A system parameter may also be modified when using the non-associated Mohr-Coulomb model.

   QMHDLM (default=0.01)

   To prevent solution instabilities a lower positive limit is applied on the hardening moduli used to form the D matrix. The default value is set at E/100 by may be altered using the system parameter QMHDLM. The value QMHDLM*E will then be used.



Cohesion Definition for the Non-Associated Mohr-Coulomb Model (Model 65)



Friction Angle Definition for the Non-Associated Mohr-Coulomb Model (Model 65)

Elasto-Plastic Stress Resultant Model

The elasto-plastic stress resultant model may be used for beams, plates and shells (see the *LUSAS Element Reference Manual*). It is based on a von Mises yield criterion. The data section MATERIAL PROPERTIES NONLINEAR 29 is used to define the material properties for the elasto-plastic stress resultant model.

```
MATERIAL PROPERTIES NONLINEAR 29

imat  E ν ρ α ar br T σy ifcode
```

imat	The material property identification number
E	Young's modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
T	Reference temperature
σ_y	Uniaxial yield stress
ifcode	Yield function code (refer to individual elements in the *LUSAS Element Reference Manual*)

Notes

1. Temperature dependent material properties are not applicable for this model.
2. The yield criteria, when used with beam elements, includes the effects of nonlinear torsion. Note that the effect of torsion is to uniformly shrink the yield surface.
3. The stress-strain curve is elastic/perfectly plastic.
4. The fully plastic torsional moment is constant.
5. Transverse shear distortions are neglected.
6. Plastification is an abrupt process with the whole cross-section transformed from an elastic to fully plastic stress state.
7. Updated Lagrangian (Option 54) and Eulerian (Option 167) geometric nonlinearities are not applicable with this model. The model, however, does support the total strain approach given by Total Lagrangian and Co-rotational geometric nonlinearities, Option 87 and Option 229, respectively.

User-Supplied Nonlinear Material Properties

The USER constitutive model facility allows the user-supplied constitutive routines to be used from within LUSAS. This facility provides completely general access to the LUSAS property data input via the MATERIAL PROPERTIES NONLINEAR USER and MATERIAL PROPERTIES NONLINEAR RESULTANT USER data sections and provides controlled access to the pre- and post-solution constitutive processing and nonlinear state variable output via these user-supplied subroutines.

By default these routines are supplied as empty routines with defined interfaces that are unchangeable. The externally developed code should be placed into these routines which is then linked into the LUSAS system. Source code access is available to these interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. See Solver User Interface Routines.

Since user specification of a constitutive model involves the external development of source FORTRAN code, as well as access to LUSAS code, this facility is aimed at the advanced LUSAS user.

Material Properties Nonlinear User

The required material properties input for the user-supplied constitutive model is completely general. The first 15 properties are reserved for LUSAS internal use: you only need to utilise them if other features of the program are required (e.g. dynamic and thermal analyses). The data section MATERIAL PROPERTIES NONLINEAR USER is used to define all the material parameters for the user-supplied constitutive models.

MATERIAL PROPERTIES NONLINEAR USER lptusr nprz nstat

imat E ν ρ α_x α_y α_z α_{xy} α_{yz} α_{xz} ar br hf T {θ.|.nset} spare < U_i >$_{i=1,(nprz-15)}$

lptusr A user defined material model number
nprz The total number of material input parameters provided
nstat The number of nonlinear state dependent constitutive variables
imat The material assignment reference number
E Young's modulus
ν Poisson's ratio
ρ Mass density
α_x α_y α_z α_{xy} α_{yz} α_{xz}
Coefficients of thermal expansion (see Notes)
ar Mass Rayleigh damping constant
br Stiffness Rayleigh damping constant
hf Heat fraction coefficient (see *Notes*)
T Reference temperature
θ Angle of anisotropy relative to the reference axes (degrees) Option 207 must be set (see *Notes*)
nset The CARTESIAN SET number used to define the local reference axes
spare Unused parameter at present (set = 0.0)
Ui The user-defined material parameters

Notes

1. LUSAS will check and diagnose erroneous or improbable data.
2. The number of material properties input must be equal to that specified on the data section header line (i.e. nprz). Failure to match the requested and supplied number of properties will invoke a LUSAS error message.

3. When no state variables are required (i.e. when nstat is specified as zero) a warning message will be invoked.
4. User-supplied constitutive models may be used as part of a composite element material assembly.
5. The first 15 material properties are required by LUSAS in order to perform other types of analyses: for example thermal problems require the coefficients of thermal expansion α (4th to 9th properties) and/or the temperature **T** (13th property), and a dynamic analysis requires the density ρ (3rd property) and the Rayleigh damping parameters **ar, br** (10th, 11th properties).
6. The user is required to input appropriate α values for the element type to be used and zeroes for the remainder. For example:
   - Plane stress elements: $\alpha_x\ \alpha_y\ 0\ \alpha_{xy}\ 0\ 0$
   - Plane strain and axisymmetric elements: $\alpha_x\ \alpha_y\ \alpha_z\ \alpha_{xy}\ 0\ 0$
   - Thick shell elements: $\alpha_x\ \alpha_y\ 0\ \alpha_{xy}\ \alpha_{yz}\ \alpha_{zx}$
   - Solid elements: $\alpha_x\ \alpha_y\ \alpha_z\ \alpha_{xy}\ \alpha_{yz}\ \alpha_{zx}$

   If an isotropic model is required then the input must be specified accordingly, e.g. for plane strain elements this would require $\alpha_x = \alpha_y = \alpha_z$ and $\alpha_{xy} = 0$.
7. If temperature dependent properties are input via the TABLE format T, the 13th property, must be specified so that the values can be interpolated for the actual temperatures at the Gauss points.
8. The 15 reserved properties can all be set to zero if you do not require other LUSAS facilities
9. Option 207 allows you to control how the local reference axes are to be determined; if the angle of anisotropy is determined by the angle q, Option 207 must be set, otherwise the reference axes must be determined by a CARTESIAN SET.
10. Option 179 can be set for argument verification within the user routines.
11. The heat fraction coefficient represents the fraction of plastic work which is converted to heat in a coupled analysis and takes a value between 0 and 1.
12. A user defined nonlinear material model which results in a nonsymmetric modulus matrix can only be used with the following element types: 3D continuum (excluding explicit dynamics elements), 2D continuum (excluding Fourier and explicit dynamics elements), bar elements and axisymmetric membrane elements.

Material Properties Nonlinear Resultant User

The general form of the input for this chapter has been tailored to allow the specification of nonlinear moment-curvature curves. However, the parameters required for any other type of user defined nonlinear resultant model may be specified via this data chapter. The first 10 properties are reserved for LUSAS internal use: you only

need to utilise some of them if other features of the program are required (e.g. dynamic and thermal analyses). The user subroutines supplied contain code that defines the moment-curvature relationship to be a function of the axial force in the member. The code in these routines can be overwritten with user defined code or alternatives added by utilising different **lptusr** parameters.

MATERIAL PROPERTIES NONLINEAR RESULTANT USER lptusr
nprz ndcrve nstat

imat E ν ρ α a_r b_r spare T spare spare …

$F_1 < M_{1,j}, C_{1,j} >_{j=1,(nprz-1)}$ …
. …
. …
. …
$F_{ndcrve} < M_{ndcrve,j}, C_{ndcrve,j} >_{j=1,(nprz-1)}$

lptusr	A user defined material model number
nprz	The number of material input parameters for each curve
ndcrve	The number of material data curves defined
nstat	The number of nonlinear state dependent constitutive variables
imat	The material assignment reference number
E	Young's modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
spare	Unused parameter at present (set = 0.0)
T	Reference temperature
spare	Unused parameter at present (set = 0.0)
spare	Unused parameter at present (set = 0.0)
F_n	Axial force for n^{th} curve (could be +ve, -ve or zero)
$M_{n,j}, C_{n,j}$	Moment and curvature for point j on the n^{th} curve

Notes

1. LUSAS will check and diagnose erroneous or improbable data.
2. The number of material data curves defined must equal the number specified, ndcrve.
3. All data curves must be defined by the same number of parameters which must equal the number specified, **nprz-1**.
4. Some of the first 10 material properties are required by LUSAS in order to perform other types of analyses: for example thermal problems require the coefficients of

thermal expansion α (4th property) and/or the temperature **T** (8th property), and a dynamic analysis requires the density ρ (3rd property) and the Rayleigh damping parameters **ar, br** (5th and 6th properties).

5. When no state variables are required (i.e. when nstat is specified as zero) a warning message will be invoked.
6. Option 179 can be set for argument verification within the user routines.
7. If temperature dependent properties are input via the TABLE format T, the 8th property, must be specified so that the values can be interpolated for the actual temperatures at the Gauss points.

Further Notes

These notes apply if the MATERIAL PROPERTIES NONLINEAR RESULTANT USER subroutines are used with the nonlinear moment-curvature facility as supplied:

1. It is recommended that the curvature values used to define the points on each individual curve be defined reasonably consistently for all curves, i.e. the curvature range used to define point j in all n curves should be reasonably small. This will lead to better interpolation between curves.
2. For 3D beam elements, all curves relating to Iyy must be specified first followed by the curves for Izz. The same number of curves must be specified for both Iyy and Izz.
3. All moment and curvature values specified for a curve must be positive. The slope of the curve segments must always be greater than zero.
4. If the computed axial force is outside the bounds of the forces defined for the data curves, the curve relating to the maximum (or minimum) axial force will be used and a warning message printed to the output file.
5. If the computed curvature exceeds the maximum value specified in the data curves, the last section of the curve will be used to compute the bending moment and a warning message will be printed to the output file.

Viscous Definition

Nonlinear viscous behaviour occurs when the relationship between stress and strain is time dependent. The viscous response is usually a function of the material together with the stress, strain and temperature history. Unlike time independent plasticity where a limited set of yield criteria may be applied to many materials, the creep response differs greatly for many materials.

To provide for the analysis of particular materials, user defined creep laws and viscoelastic models may be specified by replacing the CREEP PROPERTIES and VISCO ELASTIC PROPERTIES data sections with CREEP PROPERTIES USER and VISCO ELASTIC PROPERTIES USER respectively.

Viscous Definition

This data chapter contains the input for creep and viscoelastic material models.

VISCOUS DEFINITION

STRESS POTENTIAL {VON_MISES | HILL}

ipls [σ_{xx} σ_{yy} σ_{zz} σ_{xy} σ_{yz} σ_{xz} T]

CREEP PROPERTIES lctp

icrp $\langle f_i \rangle_{i=1,ncprp}$ T

CREEP PROPERTIES USER lctp nprzc nstat

icrp T spare spare $\langle f_i \rangle_{i=1,nprzc-3}$

VISCO ELASTIC PROPERTIES [1]

ivse G_v β T

For further information see:

- Stress Potential
- Creep Properties
- Creep Properties User
- Visco Elastic Properties
- Visco Elastic Properties User

Stress Potential

The definition of creep properties requires that the shape of the yield surface is defined (see the *LUSAS Theory Manual*). The stresses defining the yield surface are specified under the STRESS POTENTIAL command and are assigned to appropriate elements through the MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL data chapters. The STRESS POTENTIAL should only be defined under VISCOUS DEFINITION if linear material properties are to be used, otherwise, it should be defined under the PLASTIC DEFINITION data chapter. Note that the Hoffman potential is not applicable if the STRESS POTENTIAL is specified under VISCOUS DEFINITION.

STRESS POTENTIAL {VON_MISES | HILL}

ipls [σ_{xx} σ_{yy} σ_{zz} σ_{xy} σ_{yz} σ_{xz} T]

ipls	The stress potential set identification number
σij	Stresses defining the yield surface (Hill)
T	Reference temperature

Notes

1. If a stress potential type is not specified then von Mises is set as default.
2. The stress potential must be defined in full irrespective of the analysis type, except for the von Mises stress potential which, being isotropic, does not require the input of any parameter.
3. None of the stresses defining the stress potential may be set to zero. For example, in a plane stress analysis, the out of plane direct stress must be given a value which physically represents the model to be analysed.
4. STRESS POTENTIAL HOFFMAN is not applicable within the VISCOUS DEFINITION data chapter.
5. The *LUSAS Theory Manual* should be consulted if further information on these stress potential parameters is required.
6. The STRESS POTENTIAL data section can be specified under the PLASTIC DEFINITION or VISCOUS DEFINITION data chapters. If specified under the VISCOUS DEFINITION data chapter, the material properties must be linear and the STRESS POTENTIAL parameters are applied to the creep properties only. If creep is defined together with a nonlinear material property, the STRESS POTENTIAL parameters will be applied to both the creep and nonlinear material property; in this instance, the parameters must only be specified under the PLASTIC DEFINITION data chapter.
7. The STRESS POTENTIAL data is combined with other material data to define an elasto-plastic and/or a creep material model within the MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL data chapters.

Creep Properties

There are three uniaxial creep laws available in LUSAS and a time hardening form is available for all laws. The power creep law is also available in a strain hardening form. Fully 3D creep strains are computed using the differential of the von Mises or Hill stress potential. The CREEP PROPERTIES data section is used to describe the creep data for these models.

```
CREEP PROPERTIES      lctp
icrp      <fi>i=1,ncprp    T
```

lctp The creep model type:
- 1 - Power law (time dependent form)
- 2 - Power law (strain hardening form)
- 3 - Exponential law
- 4 - Eight parameter law

icrp The creep property identification number

fi Creep properties

ncprp Number of parameters defining the creep law:
- 3 - Power law
- 6 - Exponential law
- 8 - Eight parameter law

T Temperature

Example. Creep Properties

```
MATERIAL PROPERTIES
 1  2E5  0.3
VISCOUS DEFINITION
 STRESS POTENTIAL VON_MISES
  23
 CREEP PROPERTIES  1
  100 1E-7  5  0.5
MATERIAL ASSIGNMENTS
80  1  1  0  23  100
```

Notes

1. The required creep properties for each law are:

- Power law

$$\varepsilon_c = f_1 q^{f2} t^{f3}$$

- Exponential law

$$\varepsilon_c = f_1 e^{f_2 q}\left[1 - e^{-f_3 t q^{-f_4}}\right] + f_5 t_e^{f_6 q}$$

- Eight parameter law

$$\varepsilon_c = f_1 q^{f_2}\ t^{f_3} + f_4 t^{f_5} + f_6 t^{f_7}\ e^{-f_8/T}$$

where:

ε_c = uniaxial creep strain
q = von Mises or Hill equivalent deviatoric stress
t = current time
T = temperature (Kelvin)

Further information on these creep laws may be found in the *LUSAS Theory Manual*.

2. Creep properties must be defined under the VISCOUS DEFINITION data chapter

3. The definition of creep properties requires that the shape of the yield surface is defined (see the LUSAS Theory Manual). The stresses defining the yield surface are specified under the STRESS POTENTIAL command and are assigned to appropriate elements through the MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL data chapters.
4. If combined plasticity and creep is utilised then the creep and plasticity must adopt the same form of stress potential i.e. either isotropic or anisotropic.
5. Creep properties may be combined with other material properties and damage properties under the MATERIAL ASSIGNMENT or COMPOSITE MATERIAL data chapters.
6. Creep data is sometimes provided for the creep law in rate form. The time component of the law must be integrated so that the law takes a total form before data input. For example the rate form of the Power law

$$\varepsilon_c = Aq^n t^m$$

integrates to

$$\varepsilon_c = \frac{A}{(m+1)q^n t^{m+1}}$$

The properties specified as input data then become

$$f_1 = \frac{A}{m+1}$$

$$f_2 = n$$

$$f_3 = m+1$$

where A, n and m are temperature dependent constants.

7. NONLINEAR CONTROL must be specified with creep materials unless explicit integration and linear materials are specified (see Viscous Control).
8. DYNAMIC CONTROL may be utilised with creep properties if required.

User Supplied Creep Properties

The USER creep property facility allows user supplied creep law routines to be used from within LUSAS. This facility provides completely general access to the LUSAS property data input via the CREEP PROPERTIES USER data section and provides controlled access to the pre- and post-solution constitutive processing and nonlinear state variable output via these user-supplied subroutines.

CREEP PROPERTIES USER must replace CREEP PROPERTIES within the VISCOUS DEFINITION data chapter. The appropriate STRESS POTENTIAL must also be specified under VISCOUS DEFINITION, if a linear material is to be used, or

the PLASTIC DEFINITION data chapter if an allowable nonlinear material property is defined.

Source code access is available to interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. Contact FEA for full details of this facility. Since user specification of a creep law involves the external development of source FORTRAN code, as well as access to LUSAS code, this facility is aimed at the advanced LUSAS user.

CREEP PROPERTIES USER lctp nprzc nstat

icrp T spare spare < f_i >$_{i=1,nprzc-3}$

lctp	A user defined creep model type
nprzc	The number of properties for the creep model
nstat	The number of creep state variables (see *Notes*)
icrp	The creep property identification number
T	Reference temperature
spare	Unused parameter at present (set=0.0)
f_i	Creep properties

Notes

1. **nstat** must be an integer value greater than zero.
2. The number of creep properties input must be equal to that specified on the data section header line (i.e. **nprzc**). Failure to match the requested and supplied number of properties will invoke a LUSAS error message.
3. If temperature dependent properties are input using the TABLE format, **T**, the 1st property must be specified so that the values can be interpolated for the actual temperatures at the Gauss points. If the creep properties are not temperature dependent, the 1st property may be set to zero.
4. The user-supplied subroutine permits creep laws defined as:

   $$\dot{\varepsilon}_c = f(q, t, T)$$

   where

$\dot{\varepsilon}_c$	= rate of uniaxial. equivalent creep strain
q	= equivalent deviatoric stress
t	= time
T	= temperature
5. The user-supplied routine must return the increment in creep strain. Further, if implicit integration is to be used then the variation of the creep strain increment with respect to the equivalent stress and also with respect to the creep strain increment, must also be defined.
6. If the function involves time dependent state variables they must be integrated in the user-supplied routine.

7. If both plasticity and creep are defined for a material, the creep strains will be processed during the plastic strain update. Stresses in the user routine may therefore exceed the yield stress.
8. User-supplied creep laws may be used as part of a composite element material assembly.
9. Option 179 can be set for argument verification within the user routines
10. Viscoelastic properties must be defined under the VISCOUS DEFINITION data chapter.
11. The definition of creep properties requires that the shape of the yield surface is defined (see the *LUSAS Theory Manual*). The stresses defining the yield surface are specified under the STRESS POTENTIAL command and are assigned to appropriate elements through the MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL data chapters.
12. If combined plasticity and creep is utilised then the creep and plasticity must adopt the same form of stress potential i.e. either isotropic or anisotropic.
13. Creep properties may be combined with other material properties and damage properties under the MATERIAL ASSIGNMENT or COMPOSITE MATERIAL data chapters.

Visco Elastic Properties

The viscoelastic facility can be coupled with the linear elastic and non-linear plasticity, creep and damage models currently available in LUSAS. The model restricts the viscoelastic effects to the deviatoric component of the material response. This enables the viscoelastic material behaviour to be represented by a viscoelastic shear modulus G_v and a decay constant β. Viscoelasticity imposed in this way acts like a spring-damper in parallel with the elastic-plastic, damage and creep response. Coupling of the viscoelastic and the existing nonlinear material behaviour enables hysteresis effects to be modelled.

There is currently one viscoelastic model implemented in LUSAS. The VISCO ELASTIC PROPERTIES data section is used to describe the viscoelastic data for this model.

```
VISCO ELASTIC PROPERTIES      [1]
ivse  Gv  β  T
```

ivse	The viscoelastic property identification number
Gv	Viscoelastic shear modulus (see Notes)
β	Viscoelastic decay constant (see Notes)
T	Reference temperature

Notes

1. It is assumed that the viscoelastic effects are restricted to the deviatoric component of the material response. The deviatoric viscoelastic components of stress are obtained using a stress relaxation function G(t), which is assumed to be dependent on the viscoelastic shear modulus and the decay constant.

$$\sigma'_v(t) = \int_0^t 2G(t-s)\frac{d\varepsilon'}{ds}ds \qquad G(t) = G_v e^{-t\beta}$$

2. The viscoelastic shear modulus G_v can be related to the instantaneous shear modulus, G0, and long term shear modulus, G_0, using .$G_v = G_0 - G_\infty$

3. When viscoelastic properties are combined with isotropic elastic properties, the elastic modulus and Poisson's ratio relate to the long term behaviour of the material, that is, E_∞ and ν_∞ .At each iteration, the current deviatoric viscoelastic stresses are added to the current elastic stresses. The deviatoric viscoelastic stresses are updated using;

$$\sigma'_v(t+\Delta t) = \sigma'_v(t)e^{-\beta\Delta t} + 2G_v\frac{\left(1-e^{-\beta\Delta t}\right)}{\beta}\frac{\Delta\varepsilon'}{\Delta t}$$

where

σ_v' = deviatoric viscoelastic stresses
G_v = viscoelastic shear modulus
β = viscoelastic decay constant
Δt = current time step increment
$\Delta\varepsilon'$ = incremental deviatoric strains

4. When viscoelastic properties are coupled with a nonlinear material model it is assumed that the resulting viscoelastic stresses play no part in causing the material to yield and no part in any damage or creep calculations. Consequently the viscoelastic stresses are stored separately and deducted from the total stress vector at each iteration prior to any plasticity, creep or damage computations. Note that this applies to both implicit and explicit integration of the creep equations.

5. Viscoelastic properties must be defined under the VISCOUS DEFINITION data chapter.

6. Viscoelastic properties may be combined with other material properties, creep and damage properties under the MATERIAL ASSIGNMENT and COMPOSITE MATERIAL data chapters.

7. NONLINEAR CONTROL must always be specified when viscoelastic properties are assigned. In addition either DYNAMIC CONTROL or VISCOUS CONTROL must also be specified to provide a time step increment for use in the viscoelastic

constitutive equations. If no time control is used the viscoelastic properties will be ignored.

User Supplied Viscoelastic Properties

The VISCO ELASTIC PROPERTIES USER facility enables a user supplied viscoelastic model to be invoked from within LUSAS. This facility provides completely general access to the LUSAS property data input via this data section and provides controlled access to the pre- and post-solution constitutive processing and nonlinear state variable output via these user supplied routines. VISCO ELASTIC PROPERTIES USER must replace VISCO ELASTIC PROPERTIES in the VISCOUS DEFINITION data chapter.

Source code access is available to interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. Contact LUSAS for full details of this facility. Since user specification of a viscoelastic model involves the external development of a FORTRAN source code, as well as access to the LUSAS code, this facility is aimed at the advanced LUSAS user.

VISCO ELASTIC PROPERTIES USER **lvse** **nprzv** **nstat**

ivse $\langle f_i \rangle_{i=1,nprzv}$ **T**

lvse	A user defined viscoelastic model type
nprzv	The number of properties for the viscoelastic model
nstat	The number of user defined viscoelastic state variables
f_i	Viscoelastic properties
T	Reference temperature

Notes

1. **nstat** must be an integer greater than or equal to zero.
2. The number of viscoelastic properties input must be equal to that specified on the data section header line (i.e. **nprzv**). Failure to match the requested and supplied number of properties will invoke a LUSAS error message.
3. If temperature dependent properties are input using the TABLE format, **T**, the last property must be specified so that the values can be interpolated for the actual temperatures at the Gauss points.
4. A viscoelastic model can be combined with any of the elastic material models and the following nonlinear models:

   Tresca (model 61)

   Mohr Coulomb (model 65)

   Drucker-Prager (model 64)

Von-Mises (model 75)

Hill

Hoffman

User Defined Nonlinear Material Model

5. VISCO ELASTIC PROPERTIES USER must be defined under the VISCOUS DEFINITION data chapter and assigned using MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL.
6. Option 179 can be set for argument verification within the user routines.
7. Viscoelastic properties may be combined with other material properties, creep and damage properties under the MATERIAL ASSIGNMENT or COMPOSITE MATERIAL data chapters.
8. The current viscoelastic stresses must be evaluated at each iteration and added to the current Gauss point stresses. These viscoelastic stresses are subsequently subtracted at the next iteration, internally within LUSAS, before any plasticity, creep or damage calculations are performed.

Damage Material

Damage is assumed to occur in a material by the initiation and growth of cavities and micro-cracks. The DAMAGE PROPERTIES data chapter allows parameters to be defined which control the initiation of damage and post damage behaviour. In LUSAS a scalar damage variable is used in the degradation of the elastic modulus matrix. This means that the effect of damage is considered to be non-directional or isotropic. Two LUSAS damage models are available (Simo and Oliver) together with a facility for a user-supplied model. For further details of these damage models the *LUSAS Theory Manual* should be consulted.

Damage Properties

```
DAMAGE PROPERTIES  [SIMO | OLIVER]
idam  {r0 A B | r0 A n} T
```

idam	Damage properties set identification number
r0	Initial damage threshold (see *Notes*)
A	Characteristic material parameter (see *Notes*)
B	Characteristic material parameter (see *Notes*)
n	Ratio of the stresses that cause initial damage in tension and compression = **σcd**/**σtd** (see *Notes*)
T	Reference temperature

Notes

1. The initial damage threshold, **r0**, can be considered to carry out a similar function to the initial yield stress in an analysis involving an elasto-plastic material.

However, in a damage analysis, the value of the damage threshold influences the degradation of the elastic modulus matrix. A value for **r0** may be obtained from:

$$r_o = \frac{\sigma^{t}{}_{d}}{(E_0)^{1/2}}$$

where σ^{t}_{d} is the uniaxial tensile stress at which damage commences and $\mathbf{E}_o$ is the undamaged Young's modulus. The damage criterion is enforced by computing the elastic complementary energy function as damage progresses:

$$\beta\left(\sigma^{T} D_e \sigma\right)^{1/2} - r_t \leq 0$$

where σ is the vector of stress components, $\mathbf{D}_e$ the elastic modulus matrix and $\mathbf{r}_t$ the current damage norm. The factor β is taken as 1 for the Simo damage model, while for the Oliver model takes the value:

$$\beta = \left(\theta + \frac{1-\theta}{n}\right)$$

where

$$\theta = \frac{<\sigma_1> + <\sigma 2> + <\sigma 3>}{|\sigma_1| + |\sigma 2| + |\sigma 3|} \qquad n = \frac{\sigma^{d} c}{\sigma^{t} d}$$

Only positive values are considered for <**σi**>, any negative components are set to zero. The values σ^{c}_{d} and σ^{t}_{d} represent the stresses that cause initial damage in compression and tension respectively (note that if $\sigma^{c}_{d} = \sigma^{t}_{d}$, β=1). The damage accumulation functions for each model are given by:

Simo: $$G(r_t) = 1 - \frac{r_{0(1-A)}}{r_t} - A \exp\, B(r_0 - r_t)$$

Oliver: $$G(r_t) = 1 - \frac{r_0}{r_t} \exp\left[A\left(1 - \frac{r_0}{r_t}\right)\right]$$

For no damage, $\mathbf{G(r_t)=0}$. The characteristic **t** material parameters, A and B, would generally be obtained from experimental data. However, a means of computing A has been postulated for the Oliver model:

$$A = \left[\frac{G_f E_0}{l_{ch}\left(\sigma^{t}_{d}\right)^2 - \frac{1}{2}}\right]^{-1}$$

where $\mathbf{G_f}$ is the fracture energy per unit area, $\mathbf{l_{ch}}$ is a characteristic length of the finite element which can be approximated by the square root of the element area.

These damage models are explained in greater detail in the LUSAS Theory Manual.

2. The damage criterion for the Oliver model introduces a factor which is invoked if different stress levels cause initial damage in tension and compression.
3. A damage analysis can be carried out using any of the elastic material models and the following nonlinear models:
   - von Mises
   - Hill
   - Hoffman
4. CREEP PROPERTIES and/or VISCO ELASTIC PROPERTIES may be included in a damage analysis. See Viscous Definition
5. DAMAGE PROPERTIES must be assigned using MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL.

User Supplied Damage Properties

The DAMAGE PROPERTIES USER facility allows routines for implementing a user supplied damage model to be invoked from within LUSAS. This facility provides completely general access to the LUSAS property data input via this data section and provides controlled access to the pre- and post-solution constitutive processing and nonlinear state variable output via these user-supplied subroutines.

Source code access is available to interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. Contact LUSAS for full details of this facility. Since user specification of a damage model involves the external development of source FORTRAN code, as well as access to LUSAS code, this facility is aimed at the advanced LUSAS user.

DAMAGE PROPERTIES USER ldtp nprzd nstat

idam < P_i >$_{i=1,nprzd}$ T

ldtp	User defined damage model identification number
nprzd	Number of parameters used in defining the damage model
nstat	Number of damage state variables (see *Notes*)
idam	Damage properties set identification number
Pi	User supplied parameters for damage model
T	Reference temperature

Notes

1. **nstat** must be an integer value greater than zero.
2. The number of damage properties input must be equal to that specified on the data section header line (i.e. **nprzd**). Failure to match the requested and supplied number of properties will invoke a LUSAS error message.

3. If temperature dependent properties are input using the TABLE format, **T**, the last property must be specified so that the values can be interpolated for the actual temperatures at the Gauss points.
4. A damage analysis can be carried out using any of the elastic material models and the following nonlinear models:
    - von Mises
    - Hill
    - Hoffman
5. CREEP PROPERTIES and/or VISCOELASTIC PROPERTIES may be included in a damage analysis. See Viscous Definition
6. DAMAGE PROPERTIES USER must be assigned using MATERIAL ASSIGNMENTS or COMPOSITE MATERIAL.
7. Option 179 can be set for argument verification within the user routines
8. Damage properties may be combined with other material properties and creep properties under the MATERIAL ASSIGNMENT or COMPOSITE MATERIAL data chapters.

Multi-Crack Concrete Model

This concrete model is a plastic-damage-contact model in which damage planes form according to a principal stress criterion and then develop as embedded rough contact planes. The basic softening curve used in the model may be controlled via a fixed softening curve or a fracture-energy controlled softening curve that depends on the element size. The former, distributed fracture model, is applicable to reinforced concrete applications, while the latter localised fracture model is applicable to un-reinforced cases.

MATERIAL PROPERTIES NONLINEAR 94

imat E, ν, ρ, α_t, a_r, b_r, T, f_c, f_t, ε_c, ε_0, G_f, β_r ,Z_0, ψ, m_g, m_{hi}, m_{ful}, r_σ, μ, α_d

imat	The material property identification
E	Young's modulus. Only
ν	Poisson's ratio.
ρ	Mass density
αt	Coefficient of thermal expansion
ar	Mass Rayleigh damping coefficient
br	Stiffness Rayleigh damping coefficient
T	Reference temperature
fc	Uniaxial compressive strength (e.g. 40N/mm^2)
ft	Uniaxial tensile strength (e.g. 3.0 N/mm^2)
εc	Strain at peak uniaxial compression (e.g. 0.0022)

εo	Strain at effective end of softening curve for distributed fracture (e.g. 0.0035, or 0.0 if $G_f \neq 0$)
Gf	Fracture energy per unit area (e.g. 0.1 N/mm or 0.0 if $\varepsilon_0 \neq 0$)
βr	Biaxial to uniaxial peak principal stress ratio (e.g. 1.15 *Range = 1.0 to 1.25*)
Zo	Initial relative position of yield surface (e.g. 0.6. *Range = 0.1 to 1.0*)
ψ	Dilatancy factor giving plastic potential slope relative to that of yield surface (e.g. -0.1 *Range -0.25 to 1.0*)
mg	Constant in interlock state function (e.g. 0.425 *Range 0.3 to 0.6*)
mhi	Contact multiplier on ε_0 for 1st opening stage (e.g. 0.5 *Range 0.25 to 2.0*)
mful	Final contact multiplier on ε_0 (e.g. 5.0 *Range 1.0 to 20*)
rσ	Shear intercept to tensile strength ratio for local damage surface (e.g. 1.25 *Range 0.5 to 2.5*)
μ	Slope of friction asymptote for local damage surface (e.g. 1.0 *Range 0.5 to 1.5* Note $\mu < r_\sigma$)
αd	Angular limit between crack planes (e.g. 1.0 (radians))

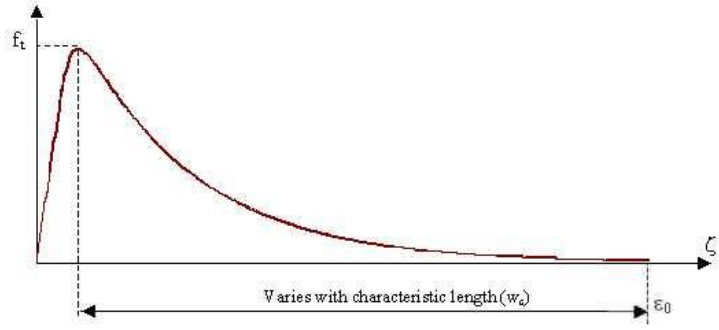
Notes

1. The model can be used with 2D and 3D continuum elements, 2D and 3D explicit dynamics elements, solid composite elements and semiloof or thick shell elements.
1. All stresses and strains should be entered as positive values.
2. If no data for the strain at peak compressive stress, ε_c, is available it can be estimated from $\varepsilon_c = 0.002 + 0.001\dfrac{(f_{cu} - 15)}{45}$ where $f_{cu} = 1.25 f_c$. Any value for ε_c should lie in the range $0.002 \le \varepsilon_c \le 0.003$. As a guide, a reasonable value for most concretes is 0.0022.
3. It is important that the initial Young's modulus, E, is consistent with the strain at peak compressive stress, ε_c. A reasonable check is to ensure that $E > 1.2(f_c / \varepsilon_c)$.
4. For concrete that contains reinforcement, distributed fracture will be the dominant fracture state. In this case a value for the strain at the end of the tensile softening curve, ε_0, should be entered and G_f set to zero. If no data is available then a value of $\varepsilon_0 = 0.0035$ is reasonable to use for most concretes.
5. For unreinforced concrete the strains will tend to localise in crack zones, leading to localised fracture. The value for ε_0 must be set to 0.0 and the fracture energy per unit area, G_f, given a positive value. G_f varies with aggregate size but not so much with concrete strength. Typical values for various maximum coarse aggregate sizes are:

   16 mm aggregate: G_f = 0.1N/mm

   20 mm aggregate: G_f = 0.13N/mm

32 mm aggregate: G_f = 0.16N/mm



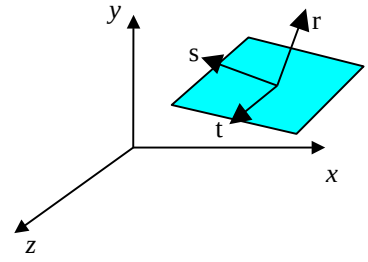
Damage evolution function

If the effective end of the softening curve parameter, ε_0, is set to zero, it will be calculated from $\varepsilon_0 \approx 5G_f / W_c f_t$ where W_c is a characteristic length for the element; if a finite value is given for ε_0, G_f will be ignored.

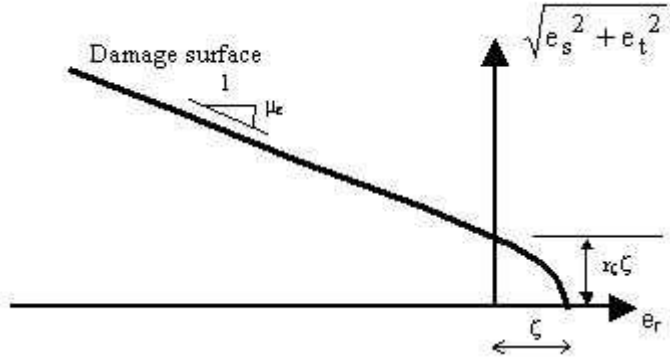
6. The initial position of the yield surface is governed by the value of Z_0. For most situations in which the degree of triaxial confinement is relatively low, a value of between 0.5 and 0.6 is considered appropriate for Z_0 however, for higher confinements a lower value of 0.25 is better.
7. The parameter ψ is used to control the degree of dilatancy. Associated plastic flow is achieved if ψ=1, but it was found that ψ values in the range -0.1 to –0.3 were required to match experimental results. Generally ψ is set to –0.1, but for high degrees of triaxial confinement –0.3 provides a better match to experimental data.
8. The constant m_g can be obtained from experimental data from tests in which shear is applied to an open crack. The default value for m_g is taken as 0.425 but it is considered that a reasonable range for m_g for normal strength concrete is between 0.3 and 0.6. However, it was found that a low value of 0.3 could lead to second cracks forming at shallow angles to the first, due to the development of relatively large shear forces.
9. It is assumed that there is a crack opening strain beyond which no further contact can take place in shear, $e_{ful.}$, where e_{ful} is a multiple of ε_0, i.e. $e_{ful}=m_{ful}\ \varepsilon_0$. Trials suggest that when concrete contains relatively large coarse aggregate i.e. 20 to 30mm, a value of m_{ful} in the range 10-20 is appropriate, whereas for concrete with relatively small coarse aggregate, i.e. 5 to 8mm, a lower value is appropriate, in the range 3 to 5. This variation is necessary because the relative displacement at the end of a tension-softening curve (related via the characteristic dimension to ε_0) is not in direct proportion to the coarse aggregate size, whereas the clearance

displacement is roughly in proportion to the coarse aggregate size. Thus e_{ful} is not in a fixed ratio to ε_0.

10. A POD is formed when the principal stress reaches the fracture stress (f_t); the POD is formed normal to the major principal axis. Thereafter, it is assumed that damage on the plane can occur with both shear and normal strains.



POD Local and Global co-ordinate systems



Local damage surface

The constants r_ζ and μ_ε are the strain equivalents of the material input parameters r_σ and μ. The relative shear stress intercept to tensile strength ratio $r_\sigma = c / f_t$ where c is the shear stress intercept.

11. Fine integration and the non-symmetric solver are always set by default with this material model.

12. It is recommended that the following LUSAS options are used with this model:

252 Suppress pivot warnings
62 Allow negative pivots

Composite Matrix Failure Model

The Hashin composite damage model has been implemented within LUSAS to model matrix/fibre failure in composite materials. The model can be used with HX16C & PN12C elements and linear material models MATERIAL PROPERTIES or MATERIAL PROPERTIES ORTHOTROPIC SOLID. A set of failure criteria have been used to represent fibre and matrix failure. These failure criteria result in a degradation of the Young's modulus, shear modulus and Poisson's ratio where the damage has occurred.

```
DAMAGE PROPERTIES HASHIN
imat      Sxt    Sxc    Sxy    Syt    Syc    T
```

where

imat	Assignment reference number
Sxt	Ply tensile strength in fibre direction
Sxc	Ply compressive strength in fibre direction
Sxy	Ply shear strength measured from a cross ply laminate
Syt	Ply transverse (normal to fibre direction) tensile strength
Syc	Ply transverse compressive strength
T	Reference Temperature

Notes

1. This damage model can only be used with the solid composite elements HX16C and PN12C
2. This model may only be used with linear material types MATERIAL PROPERTIES and MATERIAL PROPERTIES ORTHOTROPIC SOLID.
3. Nonlinear material properties and creep cannot be used with this damage model.
4. All strength values specified are positive.

Two-Phase Material

The data chapter TWO PHASE MATERIAL allows the input of two-phase material properties.

```
TWO PHASE MATERIAL
iptm Ks Kf n γ kx ky kz T
```

iptm	The two-phase material identification number
Ks	Bulk modulus of solid phase (particle)
Kf	Bulk modulus of fluid phase
n	Porosity of medium
γ	Unit weight of fluid phase

kx	Permeability in global x direction
ky	Permeability in global y direction
kz	Permeability in global z direction (not used at present)
T	Reference temperature

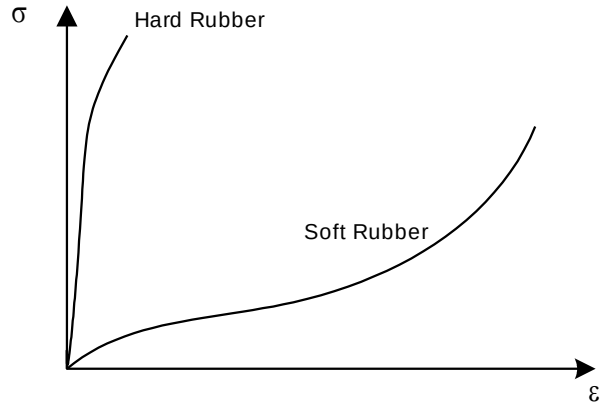
Notes

1. Usually, the value of **Ks** is quite large compared to **Kf** and not readily available to the user. If **Ks** is input as 0, LUSAS assumes an incompressible solid phase. **Kf** is more obtainable, eg. for water **Kf** = 2200 Mpa [N1]
2. Two-phase material properties can only be assigned to geotechnical elements, that is, TPN6P and QPN8P.
3. When performing a linear consolidation analysis TRANSIENT CONTROL must be specified. DYNAMIC, VISCOUS or NONLINEAR CONTROL cannot be used.
4. In an un-drained analysis two-phase material properties may be combined with any other material properties, and creep, damage and viscoelastic properties under the MATERIAL ASSIGNMENT data chapter. In a drained analysis only linear material properties may be used.
5. The two-phase material properties are assigned to the two-phase continuum elements using MATERIAL ASSIGNMENTS.

Rubber Material Properties

Rubber materials maintain a linear relationship between stress and strain up to very large strains (typically 0.1 - 0.2). The behaviour after the proportional limit is exceeded depends on the type of rubber (see diagram below). Some kinds of soft rubber continue to stretch enormously without failure. The material eventually offers increasing resistance to the load, however, and the stress-strain curve turns markedly upward prior to failure. Rubber is, therefore, an exceptional material in that it remains elastic far beyond the proportional limit.

Rubber materials are also practically incompressible, that is, they retain their original volume under deformation. This is equivalent to specifying a Poisson's ratio approaching 0.5.



Rubber Material Models

The strain measure used in LUSAS to model rubber deformation is termed a **stretch** and is measured in general terms as:

l = dnew/dold

where:

- dnew is the current length of a fibre
- dold is the original length of a fibre

Several representations of the mechanical behaviour for hyper-elastic or rubber-like materials can be used for practical applications. Within LUSAS, the usual way of defining hyper-elasticity, i.e. to associate the hyper-elastic material to the existence of a strain energy function that represents this material, is employed. There are currently four rubber material models available:

- **Ogden**
- **Mooney-Rivlin**
- **Neo-Hookean**
- **Hencky**

MATERIAL PROPERTIES RUBBER OGDEN N

imat $< \mu_{ri}\ \alpha_{ri} >_{i=1,N}$ k_r [ρ α a_r b_r T]

MATERIAL PROPERTIES RUBBER MOONEY_RIVLIN

imat C_1 C_2 k_r [ρ α a_r b_r T]

MATERIAL PROPERTIES RUBBER NEO_HOOKEAN

```
imat  C0  kr  [ρ α ar br T]

MATERIAL PROPERTIES RUBBER HENCKY

imat  G  kr  [ρ α ar br T]
```

imat	The material property identification number
μri, αri	Ogden rubber model constants
N	The number of pairs of constants for the Ogden rubber model
C1, C2	Mooney-Rivlin rubber model constants
C0	Neo-Hookean rubber model constant
G	Shear modulus
kr	Bulk modulus (see *Notes*)
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
T	Reference temperature

The constants **μr**, **αr**, **C1**, **C2** and **C0** are obtained from experimental testing or may be estimated from a stress-strain curve for the material together with a subsequent curve fitting exercise.

The Neo-Hookean and Mooney-Rivlin material models can be regarded as special cases of the more general Ogden material model. In LUSAS these models can be reformulated in terms of the Ogden model.

The strain energy functions used in these models includes both the deviatoric and volumetric parts and are, therefore, suitable to analyse rubber materials where some degree of compressibility is allowed. To enforce strict incompressibility (where the volumetric ratio equals unity), the bulk modulus tends to infinity and the resulting strain energy function only represents the deviatoric portion. This is particular useful when the material is applied in plane stress problems where full incompressibility is assumed. However, such an assumption cannot be used in plane strain or 3D analyses because numerical difficulties can occur if a very high bulk modulus is used. In these cases, a small compressibility is mandatory but this should not cause concern since only near-incompressibility needs to be ensured for most of the rubber-like materials.

Rubber is applicable for use with the following element types at present:

- 2D Continuum QPM4M, QPN4M
- 3D Continuum HX8M
- 2D Membrane BXM2, BXM3

Notes

1. For membrane and plane stress analyses, the bulk modulus kr is ignored because the formulation assumes full incompressibility. The bulk modulus has to be specified if any other 2D or 3D continuum element is used.
2. Ogden, Mooney-Rivlin and Neo-Hookean material models must be run with geometric nonlinearity using either the total Lagrangian formulation (for membrane elements) or the co-rotational formulation (for continuum elements). The Hencky material model is only available for continuum elements and must be run using the co-rotational formulation. The large strain formulation is required in order to include the incompressibility constraints into the material definition.
3. Option 39 can be specified for smoothing of stresses. This is particularly useful when the rubber model is used to analyse highly compressed plane strain or 3D continuum problems where oscillatory stresses may result in a "patchwork quilt" stress pattern. This option averages the Gauss point stresses to obtain a mean value for the element.
4. When rubber materials are utilised, the value of det F or J (the volume ratio) is output at each Gauss point. The closeness of this value to 1.0 indicates the degree of incompressibility of the rubber model used. For totally incompressible materials J=1.0. However, this is difficult to obtain due to numerical problems when a very high bulk modulus is introduced for plane strain and 3D analyses.
5. Subsequent selection of variables for displaying will include the variable PL1 which corresponds to the volume ratio.
6. Rubber material models are not applicable for use with the axisymmetric solid element QAX4M since this element does not support the co-rotational geometric nonlinear formulation. The use of total Lagrangian would not be advised as an alternative.
7. There are no associated triangular, tetrahedral or pentahedral elements for use with the rubber material models at present.
8. The rubber material models are inherently nonlinear and, hence, must be used in conjunction with the NONLINEAR CONTROL command.
9. The rubber material models may be used in conjunction with any of the other LUSAS material models.

Volumetric Crushing Model

Material behaviour can generally be described in terms of deviatoric and volumetric behaviour which combine to give the overall material response. The crushable foam material model accounts for both of these responses. The model allows you to define the volumetric behaviour of the material by means of a piece-wise linear curve of pressure versus the logarithm of relative volume. An example of such a curve is shown in the diagram below, where pressure is denoted by p and relative volume by V/V_0.

From this figure, it can also be seen that the material model permits two different unloading characteristics volumetrically.

- Unloading may be in a nonlinear elastic manner in which loading and unloading take place along the same nonlinear curve
- Volumetric crushing may be included in which unloading takes place along a straight line defined by the unloading/tensile bulk modulus K which is, in general, different from the initial compressive bulk modulus defined by the initial slope of the curve.
- Volumetric crushing is indicated by the **ivcrush** parameter.

In both cases, however, there is a maximum (or cut-off) tensile stress **pcut** that is employed to limit the amount of stress the material may sustain in tension.

The deviatoric behaviour of the material is assumed to be elastic-perfectly plastic. The plasticity is governed by a yield criterion that is dependent upon the volumetric pressure (compared with the classical von Mises yield stress dependency on equivalent plastic strain) and is defined as:

$$\tau^2 = a_0 + a_1 p + a_2 p^2$$

where p is the volumetric pressure, t is the deviatoric stress and a_0 a_1 a_2 are user defined constants. Note that, if $\mathbf{a_1} = \mathbf{a_2} = 0$ and $\mathbf{a_0} = \sigma_{yld}^2 / 3$, then classical von Mises yield criterion is obtained.

MATERIAL PROPERTIES NONLINEAR 81 N

imat K G ρ α a_r b_r h_f T pcut a_0 a_1 a_2 ivcrush < $\ln(V/V_0)_i$ p_i >$_{i=1,N}$

imat	The material property identification number
K	Bulk modulus used in tension and unloading (see figure below)
G	Shear modulus
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
hf	Heat fraction coefficient (see *Notes*)
T	Reference temperature
pcut	Cut-off pressure (see *Notes* and figure below)
a0..etc	Parameters defining pressure dependent yield stress (see *Notes*)
ivcrush	Volumetric crushing indicator (see *Notes*): 0 - no volumetric crushing 1 - volumetric crushing.

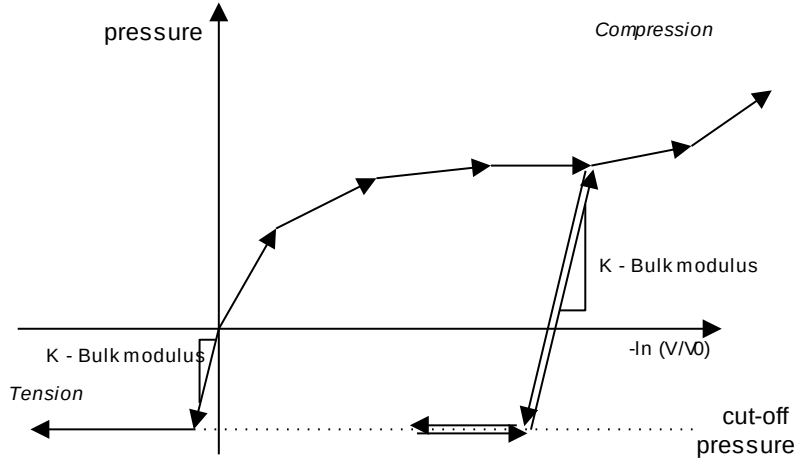
N The number of points defining the pressure-logarithm of relative volume curve in compression

ln(V/Vo)$_i$ Natural logarithm ($\log_e$, not $\log_{10}$) of relative volume coordinate for ith point on the pressure-logarithm of relative volume curve (see *Notes* and figure below)

pi Pressure coordinate for ith point on the pressure-logarithm of relative volume curve (see *Notes* and figure below)

Notes

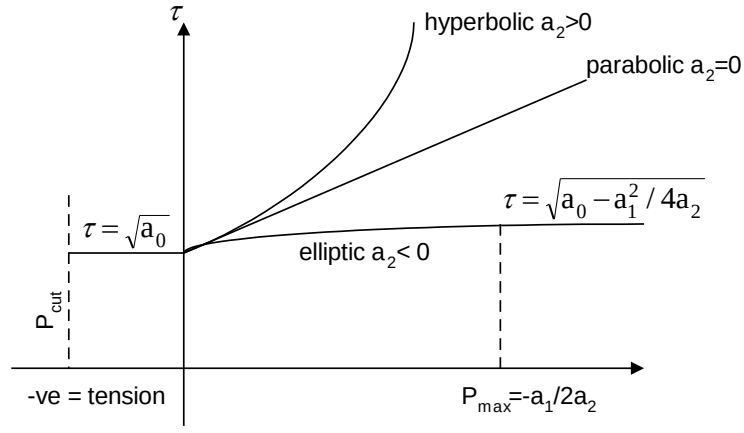
1. The heat fraction coefficient represents the fraction of plastic work which is converted to heat and takes a value between 0 and 1.
2. The pressure-logarithm of relative volume curve is defined in the compression regime hence logarithms of relative volume must all be zero or negative and the pressure coordinates must all be zero or positive.
3. The cut-off pressure should be negative (i.e. a tensile value).
4. Parameters $\mathbf{a_0}$ and $\mathbf{a_1}$ should be positive.
5. The volumetric crushing indicator effectively defines the unloading behaviour of the material. If there is no volumetric crushing, the same pressure-logarithm of relative volume curve is used in loading and unloading and if volumetric crushing takes place the alternative unloading/reloading curve is used (see figure below).



Pressure - Logarithm of Relative Volume Curve

6. The yield surface defined is quadratic with respect to the pressure variable. Therefore it can take on different conical forms (see figure below), either elliptic (a2<0), parabolic (a2=0) or hyperbolic (a2>0). The parabolic form is comparable to the modified von Mises material model while the elliptic form can be considered to be a simplification of the critical state soil and clay material behaviour. For all

values of a2 the yield criterion is taken as $\tau^2 = a_0$ when p is -ve (tension). For an elliptic surface (a2<0) the square of the yield stress is maintained at a constant maximum value (a0-a12/4a2) when the compressive pressure exceeds -a1/2a2.



Yield Surface Representation For Different a_2 Values

7. The relationship between the elastic modulus values in shear, G, and tension, E, assuming small strain conditions, is given by:

$$G = \frac{E}{2(1+\nu)}$$

8. The relationship between the elastic bulk (or volumetric) modulus, K, and tensile modulus, E, is given by:

$$K = \frac{E}{3(1-2\nu)}$$

Concrete Creep Models

LUSAS accommodates two concrete creep codes, CEB-FIP Model Code 1990 and the Chinese Creep Code for Dams. The CEB-FIP model is valid for ordinary structural concrete (12-80 Mpa) that has been loaded in compression to less than 40% of its compressive strength at the time of loading. Relative humidities in the range 40-100% and temperatures in the range 5-30 degrees C are assumed.

Although CEB-FIP Model Code 1990 is only applicable to beams, it has been extended in LUSAS to cover multi-axial stress states. The assumptions made in the derivation of this extension can be found in the LUSAS Theory Manual.

MATERIAL PROPERTIES NONLINEAR 86 [CEB-FIP | CHINESE]

imat E* ..ν, ρ, α, a_r, b_r, H_f, T, [f_r, C_t, RH, hr | a, b, f_1, g_1, p_1, r_1, f_2, g_2, p_2, r_2, f_3, r_3]

imat	The material property identification number
E*	Modulus of elasticity: CEB-FIP = E_{ci} = modulus at 28 days (see Notes) Chinese = $^{\infty}E$ = long term modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
a_r	Mass Rayleigh damping coefficient (not used)
b_r	Stiffness Rayleigh damping coefficient (not used)
H_f	Heat fraction (not used)
T	Reference temperature

CEB-FIP:

f_r	fcm/fcmo where fcm is the mean concrete compressive strength at 28 days and fcmo is a reference strength which is the equivalent of 10 Mpa in the chosen units.
C_t.	Cement type (default=2, see Notes)
RH	Relative humidity (%) of the ambient environment (default = 70%)
h_r	h/ho where h is the nominal member size (see Notes) and ho is a reference length which is the equivalent of 100mm in the chosen units.

Chinese:

a,b	Parameters for controlling evolution of elastic modulus with time (see Notes)
f_i,g_i	Parameters for controlling variation of creep coefficient with time (see Notes)
p_i,r_i	Parameters for controlling variation of creep coefficient with time (see Notes)

Notes

1. The CEB-FIP Code states that the modulus of elasticity at 28 days may be estimated from

$$E_{ci} = 2.15x10^4 x\left[\frac{f_{cm}}{10}\right]^{1/3}$$

2. In the CEB-FIP code, the cement type **C_t** is defined as:

1 for slowly hardening cements SL
2 for normal or rapid hardening cements N and R
3 for rapid hardening high strength cements RS

3. In the CEB-FIP code, the nominal size of member, **h**, is computed from $\mathbf{2A_c/u}$ where $\mathbf{A_c}$ is the area of cross section and **u** the length of the perimeter of the cross section that is in contact with the atmosphere. It should be noted that the CEB-FIP code has only been written to cover a uni-axial stress state (beams). The equations in CEB-FIP have been extended in LUSAS to cover multi-axial stress states, however, an appropriate value for **hr** must still be defined.
4. In the Chinese code, the evolution of elastic modulus with time is defined by: $^{\tau}E = {}^{\infty}E\left(1-e^{-a\tau^{b}}\right)$ where $^{\infty}E$, a and b are parameters fitted from experimental data
5. In the Chinese code, the variation of creep coefficient with time is defined by:
$$^{t,\tau}C = \left(f_1+g_1\tau^{-p_1}\right)\left[1-e^{-r_1(t-\tau)}\right]+\left(f_2+g_2\tau^{-p_2}\right)\left[1-e^{-r_2(t-\tau)}\right]+f_3e^{-r_3\tau}\left[1-e^{-r_3(t-\tau)}\right]$$
where f_i, g_i, and p_i, $r_i$are parameters fitted from experimental data.
6. In the Chinese code, parameters a, b and p_i are assumed to be dimensionless while parameters r_i are inverted retardation times and are therefore specified in days^{-1}. Parameters f_i and g_i take the units of $(^{\infty}E)^{-1}$.
7. The creep models must be run with NONLINEAR CONTROL and VISCOUS CONTROL. The time steps and total response time must be specified in days. An option exists under the INCREMENTATION chapter of VISCOUS CONTROL to use an exponent to increase the time step as the analysis progresses.
8. These material models can be combined with SHRINKAGE PROPERTIES CEB-FIP_90 to combine the effects of creep and shrinkage.

Generic Polymer Model

The polymer material model is defined using an linear spring an Eyring damper and a number of parallel Maxwell elements. Model 88 is used when behaviour differs in tension and compression.

MATERIAL PROPERTIES NONLINEAR 87 n

imat <G_i η_i >$_{i=1,n}$ A_t ν_t T G_{spr} K ρ

MATERIAL PROPERTIES NONLINEAR 88 n

imat <<G_i η_i>$_{i=1,n}$ A_t ν_t, G_{spr}>^{ten} <<G_i η_i>$_{i=1,n}$ A_t ν_t G_{spr}>>^{comp} K ρ α a_r b_r T

n	Number of Maxwell elements
imat	Material property identification number
G_i	Shear modulus of spring in *i*'th Maxwell element
η_i	Maxwell element Newtonian dashpot viscous parameter
A_t	Constant related to the activation energy of the Eyring dashpot
ν_t	Activation volume of the Eyring dashpot
T	Reference temperature
G_{spr}	Shear modulus of external spring
ten	Properties in tension
comp	Properties in compression
K	Bulk Modulus
ρ	Density
α	Coefficient of thermal expansion
a_r	Mass Rayleigh damping constant
b_r	Stiffness Rayleigh damping constant

If some of the units are not required for a particular analysis, the material parameters for these should be defined as zero.

Notes

1. **NONLINEAR CONTROL** with **VISCOUS** or **DYNAMIC CONTROL** should always be specified with this material model.
2. The number of Maxwell elements, **n**, has no restriction although the model requires at least one Maxwell element to be specified.
3. The bulk modulus, **K**, can be evaluated from values of Young's Modulus and Poisson ratio.
4. The Eyring parameters, A_t , ν_t, can be set to zero for a reduction to a linear viscoelastic material.
5. The reference temperature is for future use and has no bearing on the model.
6. The density should be specified if **DYNAMIC CONTROL** is used.
7. The external spring can be removed by setting G_{spr} to zero.

Generic Polymer Model with Damage

The polymer material model with damage consists of a linear spring, a spring that includes damage, an Eyring dashpot and a number of parallel Maxwell elements. Different sets of material properties can be defined to model tensile and compressive behaviour, except for the Visco-Scram damage model whose properties apply in both tension and compression.

Model 90 (see below) includes two additional features. The first feature is a set of failure criteria that are used to remove individual Maxwell elements when the criteria are met. The second is a switch between tensile and compressive material properties based on the stress state in an individual Maxwell element.

MATERIAL PROPERTIES NONLINEAR 89 n

imat

$\langle\langle G_i\ \eta_i\rangle_{i=1,n}\ A_t\ \nu_t\ G_{spr}\rangle^{ten}$ $\langle\langle G_i\ \eta_i\rangle_{i=1,n}\ A_t\ \nu_t\ G_{spr}\rangle^{comp}$

idam {a c K_o m V_{max} c_{max} μ_s| itdam ntpfc icdam ncpfc $\langle e_i^t\ q_i^t\rangle_{i=1,ntpfc}$ $\langle e_i^c\ q_i^c\rangle_{i=1,ncpfc}$} icrit s_u^{ten} e_u^{ten} s_u^{comp} e_u^{comp} c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8 K ρ α a_r b_r T

MATERIAL PROPERTIES NONLINEAR 90 n

imat

$\langle\langle G_i\ \tau_i\rangle_{i=1,n}\ A_t\ \nu_t\ G_{spr}\rangle^{ten}$ $\langle\langle G_i\ \tau_i\rangle_{i=1,n}\ A_t\ \nu_t\ G_{spr}\rangle^{comp}$

idam {a c K_o m V_{max} c_{max} μ_s | itdam ntpfc icdam ncpfc $\langle e_i^t\ q_i^t\rangle_{i=1,ntpfc}$ $\langle e_i^c\ q_i^c\rangle_{i=1,ncpfc}$ | a c K_o m y_{min} er_{min} y_{max} er_{max} c_{max} μ_s} icrit s_u^{ten} e_u^{ten} s_u^{comp} e_u^{comp} c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8 K ρ α a_r b_r T iswtch

n	Number of Maxwell elements
imat	Material property identification number
G_i	Shear modulus of spring in *i*'th Maxwell element
η_i	Viscous constant of Newtonian dashpot in *i*'th Maxwell element
τ_i	Retardation time of Newtonian dashpot for *i*'th Maxwell element
A_t	Constant related to the activation energy of the Eyring dashpot
ν_t	Activation volume of the Eyring dashpot
G_{spr}	Shear modulus of external spring
ten	Properties in tension
comp	Properties in compression
idam	Damage model =1 ViscoScram model with constant V_{max} =2 Strain-based damage model =3 ViscoScram model with strain rate dependant V_{max}

Visco Scram damage model with constant V_{max} (idam=1)

a	Initial flaw size
c	Average crack growth
K_o	Threshold value of stress intensity

m	Cracking parameter
V_{max}	Maximum value of rate of growth of average crack radius
c_{max}	Maximum crack length
μ_s	Static coefficient of friction

Strain-based damage model (idam=2)

itdam	Tension function identifier =1 User defined =2 No damage
ntpfc	Number of points on damage curve for tensile damage function
icdam	Compression function identifier =1 User defined =2 No damage
ncpfc	Number of points on damage curve for compressive damage function
e_i	x-coordinate of damage function
q_i	y-coordinate of damage function ("Q")

Visco Scram damage model with strain rate dependant V_{max} (idam=3)

a	Initial flaw size
c	Average crack growth
K_o	Threshold value of stress intensity
m	Cracking parameter
y_{min}	The value $\ln\left(V_{max}\left(\dot{\varepsilon}_{min}\right)\right)$ *(see notes)*
er_{min}	The value $\dot{\varepsilon}_{min}$ *(see notes)*
y_{max}	The value $\ln\left(V_{max}\left(\dot{\varepsilon}_{max}\right)\right)$ *(see notes)*
er_{max}	The value $\dot{\varepsilon}_{max}$ *(see notes)*
c_{max}	Maximum crack length
μ_s	Static coefficient of friction

Failure properties

icrit	Failure criteria = 1 Maximum shear stress criterion = 2 Von Mises criterion = 3 Maximum normal stress criterion = 4 Mohr's theory = 5 Maximum strain theory = 6 Critical strain theory
s_u^{ten}	Ultimate stress in tension
e_u^{ten}	Ultimate strain in tension
s_u^{comp}	Ultimate stress in compression
e_u^{comp}	Ultimate strain in compression
c_1	Gradient of ultimate stress vs. $\log_e$(effective strain rate) graph (tension)

c_2	Intercept of ultimate stress vs. $\log_e$(effective strain rate) graph with ultimate stress axis (tension)
c_3	Gradient of ultimate strain vs. $\log_e$(effective strain rate) graph (tension)
c_4	Intercept of ultimate strain vs. $\log_e$(effective strain rate) graph with ultimate strain axis (tension)
c_5	Gradient of ultimate stress vs. $\log_e$(effective strain rate) graph (compression)
c_6	Intercept of ultimate stress vs. $\log_e$(effective strain rate) graph with ultimate stress axis (compression)
c_7	Gradient of ultimate strain vs. $\log_e$(effective strain rate) graph (compression)
c_8	Intercept of ultimate strain vs. $\log_e$(effective strain rate) graph with ultimate strain axis (compression)

Generic properties

K	Bulk Modulus
ρ	Density
α	Coefficient of thermal expansion
a_r	Mass Rayleigh damping constant
b_r	Stiffness Rayleigh damping constant
T	Reference temperature
Iswtch	Tensile/compressive material property switch =0 switch on mean stress =1 switch on individual Maxwell element

Notes

1. **NONLINEAR CONTROL** with either **VISCOUS** or **DYNAMIC CONTROL** should always be specified when using this material model.
2. At least one Maxwell element must be specified. However, there is no upper limit to n, the number of Maxwell elements.
3. The Newtonian damper in any Maxwell element can be turned off. For Model 89 this is achieved by setting the Maxwell element's viscous constant, η_i, to zero, while for Model 90 the retardation time, τ_i, must be set to zero. The Maxwell element will then behave as a spring in parallel with the other Maxwell elements.
4. The strain rate across the Eyring dashpot is related to the Eyring parameters via the equation

   $$\dot{\varepsilon}^{eyr} = A_t \sinh(\nu_t \sigma_e)$$

   where σ_e is the effective deviatoric stress.
5. The Eyring parameters, A_t and ν_t, can be set to zero to remove the effect of the Eyring dashpot.

6. Properties for the Maxwell elements and the Eyring dashpot must be specified for both tension and compression.

7. There are two ViscoScram damage models, `idam=1,3`. With `idam=1` a constant value for V_{max} is used, whilst with `idam=3` a log-log relationship between V_{max} and the strain rate is used (see below).

8. For the Viscoscram damage model that uses a strain rate dependent V_{max} (`idam=3`), the following relationship between V_{max} and the effective strain rate, $\dot{\varepsilon}$, applies.

$$\ln\left(V_{max}\right) = \ln\left(V_{max}\left(\dot{\varepsilon}_{min}\right)\right) + \left[\frac{\ln\left(V_{max}\left(\dot{\varepsilon}_{max}\right)\right) - \ln\left(V_{max}\left(\dot{\varepsilon}_{min}\right)\right)}{\ln\left(\dot{\varepsilon}_{max}\right) - \ln\left(\dot{\varepsilon}_{min}\right)}\right]\left[\ln\left(\dot{\varepsilon}\right) - \ln\left(\dot{\varepsilon}_{min}\right)\right]$$

where

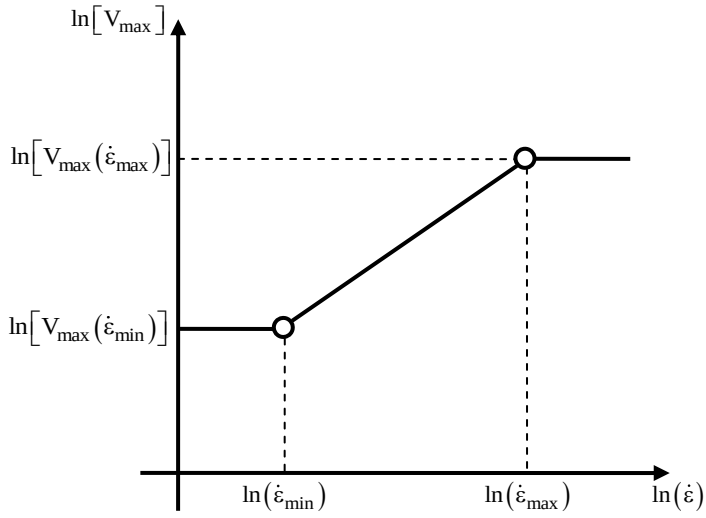
$V_{max}\left(\dot{\varepsilon}_{min}\right)$	Maximum rate of growth of the average crack radius, V_{max}, at the *minimum* valid strain rate, $\dot{\varepsilon}_{min}$
$V_{max}\left(\dot{\varepsilon}_{max}\right)$	Maximum rate of growth of the average crack radius, V_{max}, at the *maximum* valid strain rate $\dot{\varepsilon}_{max}$
$\dot{\varepsilon}_{min}$	Minimum effective strain rate at which the log-log relationship with V_{max} is valid
$\dot{\varepsilon}_{max}$	Maximum effective strain rate at which the log-log relationship with V_{max} is valid

The relationship between V_{max} and $\dot{\varepsilon}$ applies in the range $\dot{\varepsilon}_{min} < \dot{\varepsilon} < \dot{\varepsilon}_{max}$. The value of $\ln\left(V_{max}\right)$ is bounded as follows.

$\ln\left(V_{max}\right) = \ln\left(V_{max}\left(\dot{\varepsilon}_{min}\right)\right)$ for $\dot{\varepsilon} < \dot{\varepsilon}_{min}$

$\ln\left(V_{max}\right) = \ln\left(V_{max}\left(\dot{\varepsilon}_{max}\right)\right)$ for $\dot{\varepsilon} > \dot{\varepsilon}_{max}$

The relationship between V_{max} and $\dot{\varepsilon}$ is shown as a graph in the figure below.



9. The damage deviatoric strain in both ViscoScram damage models (`idam=1,3`) is defined as

$$e_{ij}^{dam} = \beta c^3 s_{ij}$$

and the damage deviatoric strain rate is defined as

$$\dot{e}_{ij}^{dam} = \frac{3}{2\bar{G}}\left(\frac{c}{a}\right)^2 \frac{\dot{c}}{a} s_{ij} + \frac{1}{\bar{G}}\left(\frac{c}{a}\right)^3 \dot{s}_{ij}$$

where $\bar{G}$ is the sum of the Maxwell shear modulii.

10. If no damage is required with the ViscoScram damage models (`idam=1,3`), the value for a, the initial flaw size, should be set to zero.

11. For the ViscoScram damage models (`idam=1,3`) only one set of damage properties is specified to cover both tensile and compressive cases.

12. With failure, if any of the terms c_1-c_8 are non-zero then the ultimate stresses and strains, s_u and e_u, in tension and compression are computed from the following equations.

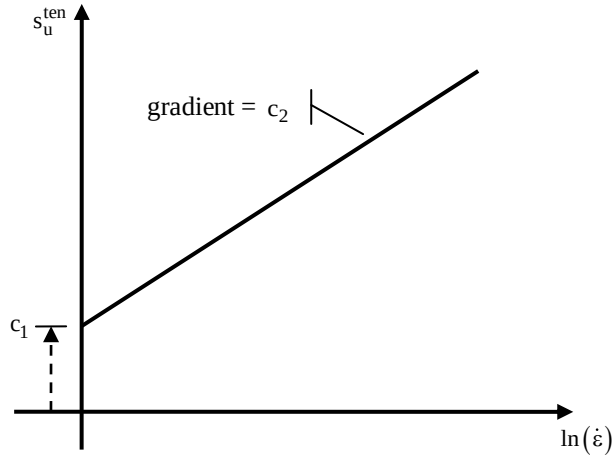
$$s_u^{ten} = c_1 \ln(\dot{\varepsilon}) + c_2 \qquad e_u^{ten} = c_3 \ln(\dot{\varepsilon}) + c_4$$

$$s_u^{comp} = c_5 \ln(\dot{\varepsilon}) + c_6 \qquad e_u^{comp} = c_7 \ln(\dot{\varepsilon}) + c_8$$

$\dot{\varepsilon}$ is the effective strain rate, `ten` indicate tensions and `comp` indicates compression. The values computed above override any specification for s_u and e_u in the data input. The sign convention for the data input is that the ultimate stresses/strains in tension are positive and in compression are negative.

The terms c_1- c_8 are determined from graphs of ultimate stress/strain vs. the natural logarithm of the strain rate in tension and compression. The figure below

shows an example for the ultimate stress in tension. The graph shows the quantities c_1 and c_2 that define the relationship between s_u^{ten} and $\ln(\dot{\varepsilon})$.



13. The bulk modulus, K, can be computed from standard isotropic relations using the Young's Modulus, E, and the Poisson ratio, ν, of the material.

$$K = \frac{E}{3(1 - 2\nu)}$$

14. The density, ρ, must be specified if DYNAMIC CONTROL is used.

15. The parameter iswtch determines when a switch between tensile and compressive material properties occurs. With iswtch=0, the criteria is based on the mean stress. With iswtch=1, each Maxwell element has its tensile/compressive material properties selected based on the stress state within that Maxwell element.

16. The units of some GPM variables can be expressed in base dimensions M (mass), L (length), T (time) and Θ (temperature).

GPM unit	Parameter	Units	Parameter	Units
Maxwell	G_i	$M.L^{-1}.T^{-2}$	τ_i	T
Eyring	A_t	T^{-1}	ν_t	$M^{-1}.L.T^{2}$
ViscoScram	a	L	V_{max}	$L.T^{-1}$
	c	L	c_{max}	L
	K_0	$M.L^{-1/2}.T^{-2}$	μ_s	Dimensionless
	m	Dimensionless		
Generic	K	$M^{-1}.L.T^{2}$	a_r	Dimensionless
	ρ	$M.L^{-3}$	b_r	Dimensionless
	α	1/Θ	T	Θ
Miscellaneous	Stress	$M.L^{-1}.T^{-2}$	Strain rate	T^{-1}

GPM unit	Parameter	Units	Parameter	Units
	Strain	Dimensionless		

2D Elasto-Plastic Interface Model

The elasto-plastic interface models may be used to represent the friction-contact relationship within planes of weakness between two discrete 2D bodies. The model is available in plane stress and plane strain elements (see the *LUSAS Element Reference Manual*), a line of which must lie between the two bodies in the finite element discretisation.

The elastic material properties are defined in the local basis, permitting differing values to be specified normal and tangential to the plane of the interface. The nonlinear behaviour is governed by an elasto-plastic constitutive law, which is formulated with a limited tension criterion normal to the interface plane, and a Mohr-Coulomb criterion tangential to the interface plane.

The data section MATERIAL PROPERTIES NONLINEAR 27 is used to define the material properties for the 2D elasto-plastic interface model.

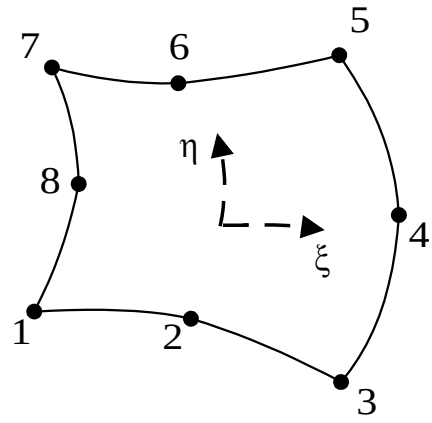
MATERIAL PROPERTIES NONLINEAR 27

imat E_{in} E_{out} **G** ν ρ α a_r b_r h_f **T** **c** ϕ σ_y

imat	The material property identification number
E_{in}	Young's modulus in-plane
E_{out}	Young's modulus out-of-plane
G	Shear modulus
ν	Poisson's ratio
ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
hf	Heat fraction coefficient (see *Notes*)
T	Reference temperature
c	Cohesion
ϕ	Friction angle
σ**y**	Uniaxial yield stress

Notes

1. The model cannot be used within a geometrically nonlinear analysis.
2. The tangential (in-plane) direction is in the element x direction and the normal (out of plane) is in the element h direction. The x direction is defined from the vector between nodes 1 and 3 in the following diagram:



3. The heat fraction coefficient represents the fraction of plastic work which is converted to heat and takes a value between 0 and 1. For compatibility with pre LUSAS 12 data files specify Option -235.

3D Elasto-Plastic Interface Model

The elasto-plastic interface models may be used to represent the friction-contact relationship within planes of weakness between two discrete 3D bodies. The model is available in solid continuum elements (see the *LUSAS Element Reference Manual*), a line of which must lie between the two bodies in the finite element discretisation.

The elastic material properties are defined in the local basis, permitting differing values to be specified normal and tangential to the plane of the interface. The nonlinear behaviour is governed by an elasto-plastic constitutive law, which is formulated with a limited tension criterion normal to the interface plane, and a Mohr-Coulomb criterion tangential to the interface plane.

The data section MATERIAL PROPERTIES NONLINEAR 26 is used to define the material properties for the 3D elasto-plastic interface model.

```
MATERIAL PROPERTIES NONLINEAR 26
```

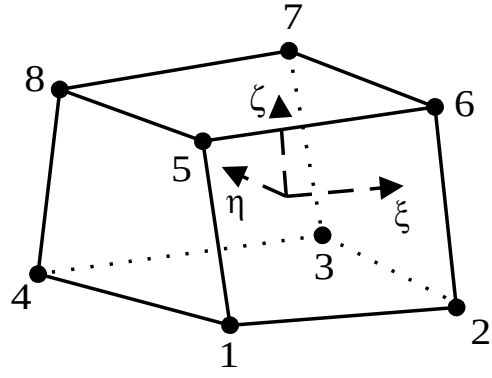
imat E_{in} E_{out} G_{in} G_{out} ν_{in} ν_{out} ρ α a_r b_r h_f **T** **c** ϕ σ_y

imat	The material property identification number
Ein	Young's modulus in-plane
Eout	Young's modulus out-of-plane
Gin	In-plane shear modulus
Gout	Out-of-plane shear modulus
vin	In-plane Poisson's ratio
vout	Out-of-plane Poisson's ratio

ρ	Mass density
α	Coefficient of thermal expansion
ar	Mass Rayleigh damping constant
br	Stiffness Rayleigh damping constant
hf	Heat fraction coefficient (see *Notes*)
T	Reference temperature
c	Cohesion
ϕ	Friction angle
σy	Uniaxial yield stress

Notes

1. The model cannot be used within a geometrically nonlinear analysis
2. The tangential (in-plane) direction lies in the element x-h plane and the normal (out of plane) in the element z direction.



3. The heat fraction coefficient represents the fraction of plastic work which is converted to heat and takes a value between 0 and 1. For compatibility with pre LUSAS 12 data files specify Option -235.

Delamination Interface Properties

The nonlinear material model 25 is used to define the delamination interface properties

MATERIAL PROPERTIES NONLINEAR 25 idim

imat (Gi σ_t ε_i mode)$_{i=1,idim}$

where

idim	number of dimensions of model (IPN6 =2, IS16 = 3)
imat	material number

G_i	critical fracture energy
σ_t	tension threshold/interface strength
ε_i	maximum relative displacement
mode	unloading model
	1 - reversible unloading
	2 - unloading on secant towards origin
	3 - coupling model

Notes

1. It is recommended that the arc length procedure is adopted with the option to select the root with the lowest residual norm OPTION 261
2. It is recommended that fine integration is selected for the parent elements using OPTION 18
3. The nonlinear convergence criteria should be selected to converge on the residual norm.
4. OPTION 62 should be selected to continue if more than one negative pivot is encountered and OPTION 252 should be used to suppress pivot warning messages from the solution process.
5. The non-symmetric solver is selected automatically when mixed mode delamination is specified.
6. The critical fracture energies should be the measured GIc, and GIIc
7. The tension threshold /interface strength is the stress at which delamination is initiated. This should be a good estimate of the actual delamination tensile strength but, for many problems the precise value has little effect on the computed response. If convergence difficulties arise it may be necessary to reduce the threshold values to obtain a solution.
8. The maximum relative displacement is used to define the stiffness of the interface before failure. Provided it is sufficiently small to simulate an initially very stiff interface it will have little effect. Typically its value should be defined as 10 -7.
9. Although the solution is largely independent of the mesh discretisation, to avoid convergence difficulties it is recommended that a least two elements are placed in the process zone.

Resin Cure Model

This model is used for predicting the deformations of thermoset composites that occur during a hot cure manufacturing process. The effects of chemical shrinkage (via SHRINKAGE PROPERTIES) and thermal expansion are accounted for along with the evolution of material properties during the cure cycle. This facility is intended for use within the framework of a semi-coupled thermo-mechanical analysis. However, a simplified solution without thermo-mechanical coupling can be computed for the situation in which the cure proceeds uniformly within a part by prescribing the resin

state plus accompanying shrinkage, at discrete points within the cure process. To fully utilise this model the High Precision Moulding product is required.

Either layer or basic fibre/resin properties can be entered. Properties are required in both the glassy and rubbery states. Additionally, properties in the liquid state are required for layer properties. The basic fibre/resin properties are fed into a micromechanics analysis to produce the corresponding layer values. Full laminate values may also be entered.

MATERIAL PROPERTIES NONLINEAR CURE LAYER [DIBENEDETTO | GENERAL | USER] [npnts] [nstate]

Imat E_{11}^G E_{22}^G E_{33}^G G_{12}^G G_{23}^G G_{13}^G ν_{12}^G ν_{23}^G ν_{13}^G α_{11}^G α_{22}^G α_{33}^G β_{11}^G β_{22}^G β_{33}^G E_{11}^R E_{22}^R E_{33}^R G_{12}^R G_{23}^R G_{13}^R ν_{12}^R ν_{23}^R ν_{13}^R α_{11}^R α_{22}^R α_{33}^R β_{11}^R β_{22}^R β_{33}^R E_{11}^L E_{22}^L E_{33}^L G_{12}^L G_{23}^L G_{13}^L ν_{12}^L ν_{23}^L ν_{13}^L α_{11}^L α_{22}^L α_{33}^L β_{11}^L β_{22}^L β_{33}^L **cGel** ρ a_r b_r **<** f_i **>** $_{i=1,npnts}$

MATERIAL PROPERTIES NONLINEAR CURE FIBRE_RESIN [DIBENEDETTO | GENERAL | USER] [npnts] [nstate]

imat E_{fib1} E_{fib2} G_{fib12} G_{fib23} ν_{fib} α_{fib1} α_{fib2} ρ_{fib} E_{mat}^G ν_{mat}^G α_{mat}^G E_{mat}^R ν_{mat}^R α_{mat}^R **cGel** ρ_{mat} a_r b_r **ftype WarpU FillU FibH FibW Wc Fc Acl nGP Atype < f_i >** $_{i=1,npnts}$

DIBENEDETTO	DiBenedetto glass transition model, see *Notes*
GENERAL	Piecewise linear description of glass transition temperature, see *Notes*
USER	User defined glass transition function, see *Notes*
$*^G$	glassy property
$*^R$	rubbery property
$*^L$	liquid property
npnts	number of data parameters which define glass transition equation (not required if a glass transition function is not declared)
nstate	number of user state variables (minimum of 1 if user model specified)
imat	The material property identification number
E_{ii}	Young's modulus of the laminar in the longitudinal, transverse and thickness directions.
E_{fib1}	Young's modulus of the fibre in the longitudinal direction
E_{fib2}	Young's modulus of the fibre in the transverse direction

E_{mat}	Young's modulus of the matrix
G_{ij}	Shear moduli for the laminar
G_{fib12}	Shear modulus along fibre
G_{fib23}	Shear modulus across fibre
G_{mat}	Shear modulus of the matrix
ν_{ij}	Poisson's ratios of the laminar
ν_{fib}	Poisson's ratio of the fibre
ν_{mat}	Poisson's ratio of the matrix
α_{ii}	Coefficients of thermal expansion in the longitudinal, transverse and thickness directions of the laminar
α_{fib1}	Coefficients of thermal expansion along the fibre
α_{fib2}	Coefficients of thermal expansion across the fibre
α_{mat}	Coefficient of thermal expansion of the matrix
β_{ii}	Coefficients of shrinkage, with respect to degree of cure, in the longitudinal, transverse and thickness directions
cGel	degree of cure at which matrix gels
ρ	Mass density of the laminar
ρ_{mat}	Mass density of the resin
a_r	Rayleigh mass damping constant
b_r	Rayleigh stiffness damping constant
ftype	Fabric type, see *Notes*
WarpU	Warp undulation fraction (recommended 1.0)
FillU	Fill undulation fraction (recommended 1.0)
FibH	Fibre height fraction of layer height (recommended 1.0)
FibW	Fibre width fraction (recommended 1.0)
Wc	Warp count, fibre bundles per unit length in the warp direction
Fc	Fill count, fibre bundles per unit length in the weft direction
Acl	Aerial density of the fabric
nGP	Number of Gauss points (1-11) (recommended 8)
Atype	Woven fabric analysis type (1 = isostrain (recommended), 2 = isostress)
< f_i >	Functional values which depend on the form of the Glass transition equation selected, see *Notes*

Notes

1. The resin cure models are intended for use with the High Precision Moulding product.
2. These models are utilised in a thermo-mechanical coupled analysis where the degree of cure is computed in the thermal analysis.
3. The shrinkage of the resin is introduced via the SHRINKAGE PROPERTIES GENERAL or USER chapters.

4. No glass transition function need be entered if the material states are defined using the STATE command in VISCOUS CONTROL.

5. The following options are available for **`ftype`**, the fabric type.

ftype	Description	ftype	Description
1	Plain	8	Satin 5H
2	Leno	9	Satin 8H
3	Mock Leno	10	Satin 12H
4	Twill 2x1	11	Basket 2x2
5	Twill 3x1	12	Basket 4x4
6	Twill 2x2	13	Basket 8x8
7	Twill 4x4	14	0/90 UD (non-crimp)
ab	UD Tape where 'a' is the mechanical model and 'b' the thermal model. For example to use the CCA model to define the mechanical properties and Chamis the thermal 'ab' = '215'		

a	Mech.Model	b	Therm.Model
1	Rule of mixtures	11	Rule of mixtures
2	CCA	12	CCA
3	Puck	13	Schneider
4	Chamis	14	Shapery
5	Halpin Tsai (circ)	15	Chamis
6	Halpin Tsai (rect)	16	Chamberlain (hexag)
		17	Chamberlain (square)

6. The functional values which depend on the form of the Glass transition equation selected are defined as follows

DiBenedetto equation, npnts=3

f_1 = TgI glass transition temperature of fully cured polymer

f_2 = Tg0 glass transition temperature before cure

lambda = material parameter

General piece-wise linear, npnts = number of points defining linear sections

$< DOC_i \ Tg_i >_{i=1,npnts}$ DOC degree of cure, Tg glass transition temperature

User equation defined via user interface, npnts=number of user input parameters

$< f_i >_{i=1,npnts}$ where

f_1 = LTGUSR – identifier passed into user routine

fi (i=2,npnts) = parameters passed to user routine

7. The User defined glass transition function is introduced via externally developed FORTRAN source code. Source code access is available to interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. Contact LUSAS for full details of this facility.
8. A minimum data solution is available which is applicable to thin composites in which the state of cure is constant or nearly constant at every point. This approach requires use of the STATE command in VISCOUS CONTROL and does not involve a coupled analysis.
9. The coupled analysis solution requires the use of NONLINEAR CONTROL in both the thermal and structural analysis. The thermal analysis also requires the TRANSIENT CONTROL chapter and the structural analysis VISCOUS CONTROL. A semi-coupled analysis with data transfer at every time step should be used.

Shrinkage

The cure of concrete and thermoset resins is accompanied by an isotropic shrinkage which in concrete's case depends on time, temperature and other environmental factors whilst for thermoset resins the shrinkage is normally described with respect to its degree of cure. LUSAS provides for the shrinkage of concrete using equations defined in the design code CEB FIP 90 and also a more general routine in which shrinkage is defined using a piecewise linear curve. Using the latter, shrinkage can be a function of either time or degree of cure. Additionally a user facility is available which provides a means of interfacing LUSAS with externally developed code.

Shrinkage Properties CEB-FIP 90

```
SHRINKAGE PROPERTIES CEB_FIP_90
ishr  Ct  RH  hr  fr
```

ishr Shrinkage property identification number

Ct Cement type (see notes)

RH Relative humidity (%) of the ambient environment

hr h/h_o where h is the nominal member size (see notes) and h_o is the equivalent of 100mm in the chosen units.

fr f_{cm}/f_{cmo} where f_{cm} is the mean concrete compressive strength at 28 days and f_{cmo} is a reference strength which is the equivalent of 10 MPa in the chosen units.

Notes

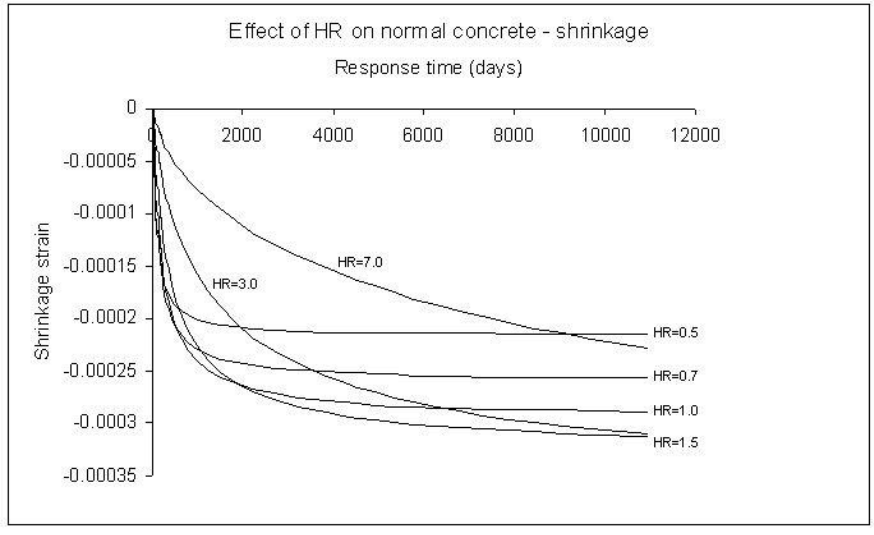
1. The cement type, **Ct**, is defined as:

1 for slowly hardening cements SL

2 for normal or rapid hardening cements N and R

3 for rapid hardening high strength cements RS

2. The nominal size of member, **h**, is computed from **2Ac/u** where **Ac** is the area of cross section and **u** the length of the perimeter of the cross section that is in contact with the atmosphere. The CEB-FIP Model Code 1990 is only strictly applicable to a uni-axial stress state but the law has been extended in LUSAS to accommodate multi-axial stress states. Care should be taken when estimating a value to use for **h** when applying CEB-FIP shrinkage to concrete members that are not beam-like in nature. In general, the larger the value for **h**, the longer the time taken for shrinkage strains to reach a final value; for large values of **h**, it must be decided whether this behaviour is reasonable. An illustration of the effect on shrinkage of varying the input parameter **hr** is shown below. As CEB-FIP creep and shrinkage input parameters can be defined separately, it is possible, if necessary, to use different **hr** values for creep and shrinkage on the same assignment.



3. The CEB-FIP shrinkage model can be used with the CEB-FIP creep model.

4. Shrinkage is calculated from the time of activation of the element.

Shrinkage Properties General

SHRINKAGE PROPERTIES GENERAL [TIME | DEGREE_OF_CURE] npnts

ishr < f_i V^s_i > $_{i=1.npnts}$

TIME Interpolate linear shrinkage using time measured from the element's activation

Degree_of_cure Interpolate linear shrinkage from degree of cure

npnts Number of points defining piecewise-linear linear shrinkage curve

fi time or degree of cure at point i

Vsi Linear shrinkage at time/degree of cure at point i.

Notes

1. The linear shrinkage value V^s_i is the shrinkage in the coordinate direction. It is applied equally in the x, y and z coordinate directions according to the particular element stress type.
2. The degree of cure is calculated from a coupled thermal analysis which takes account of the variation of temperature throughout the model.
3. Shrinkage is calculated from the time of activation of the element. That is the interpolation time is taken to be zero on activation.

Shrinkage Properties User

The SHRINKAGE PROPERTIES USER facility provides routines for implementing a user supplied shrinkage model to be invoked from within LUSAS. This facility provides completely general access to the LUSAS property data input via this data section and controlled access to the pre-solution processing and nonlinear state variable output.

Source code access is available to interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. Contact LUSAS for full details of this facility. Since user specification of a shrinkage model involves the external development of source FORTRAN code, as well as access to LUSAS code, this facility is aimed at the advanced LUSAS user.

SHRINKAGE PROPERTIES USER lstp nprzs nstat

ishr $< f_i >_{i=1,nprzs}$

lstp User defined shrinkage model identification number

nprzs Number of parameters used in definition of shrinkage model

nstat Number of shrinkage state variables (see *Notes*)

fi User supplied parameters for shrinkage model

Notes

1. **nstat** must be an integer greater than zero
2. The number of damage properties input must be equal to that specified on the data section header line (i.e. nprzs).
3. Option 179 can be set for argument verification within the user routines

Field Material Properties

The following section contains field element material properties, one of which must be defined for each field element; these material properties are not applicable to any other element type. The following types of field material properties are available:

- **Isotropic Field Model**
- **Orthotropic Plane Field Model**
- **Orthotropic Solid Field Model**

If temperature dependent material properties are specified, the linear problem becomes nonlinear and NONLINEAR CONTROL will need to be specified to obtain a correct solution. TEMP and TMPE loading may be used to provide an initial thermal field for the evaluation of temperature dependent properties. Further iterations will use the evaluated thermal field.

Note that for temperature dependent thermal boundary conditions, such as radiation, it is computationally more efficient to use thermal links to model the boundary heat transfer than to apply it directly to the elements as a load type.

Isotropic Field Model

The isotropic field model may be used to represent field properties which are identical in each direction. The data section MATERIAL PROPERTIES FIELD ISOTROPIC is used to specify the material properties for the isotropic field model.

```
MATERIAL PROPERTIES FIELD ISOTROPIC  n
imat  k T [C H]
```

imat	The material property identification number
n	Phase change type: =0 Phase change not required (default) =1 Del Giudice =2 Lemmon
k	Thermal conductivity
T	Reference temperature
C	Specific heat coefficient
H	Enthalpy

Notes

1. The specific heat coefficient(s) must be multiplied by the density prior to input.
2. Isotropic field properties are applicable to all field elements except thermal links (see the *LUSAS Element Reference Manual*).
3. It is recommended that Option 105 (lumped specific heat) is used with phase change analyses.
4. Specific heat coefficient is only required for transient analyses.

5. Enthalpy is only required for phase change analyses.

Plane Orthotropic Field Model

The data section MATERIAL PROPERTIES FIELD ORTHOTROPIC is used to define the material properties for the plane orthotropic field model. The model is valid for 2D plane field problems.

```
MATERIAL PROPERTIES FIELD ORTHOTROPIC  n
imat  kx ky θ T [C H]
```

imat	The material property identification number
n	Phase change type: =0 Phase change not required (default) =1 Del Giudice =2 Lemmon
kx, ky	Thermal conductivity in x and y directions
θ	angle of orthotropy (degrees) relative to global X-axis (positive in an anti-clockwise direction)
T	Reference temperature.
C	Specific heat coefficient (only required for transient analysis).
H	Enthalpy (only required for phase change analysis).

Notes

1. The specific heat coefficient(s) must be multiplied by the density prior to input.

2. It is recommended that Option 105 (lumped specific heat) is used with phase change analyses.

Orthotropic Solid Field Model

The data section MATERIAL PROPERTIES FIELD ORTHOTROPIC SOLID is used to define the material properties for orthotropic solid field problems. The model is valid for 3D solid field problems.

```
MATERIAL PROPERTIES FIELD ORTHOTROPIC SOLID  n
imat  nset kx ky kz T [C H]
```

imat	The material property identification number
n	Phase change type: =0 Phase change not required (default) =1 Del Giudice =2 Lemmon
nset	CARTESIAN SET number used to define local axes direction. If nset=0, the local axes coincide with global axes.
kx, ky, kz	Thermal conductivity in x, y and z directions

T Reference temperature.
C Specific heat coefficient (only required for transient analysis).
H Enthalpy (only required for phase change analysis).

Notes

1. The specific heat coefficient(s) must be multiplied by the density prior to input.
2. It is recommended that Option 105 (lumped specific heat) is used with phase change analyses.

Isotropic Concrete Heat of Hydration Model

The data section MATERIAL PROPERTIES FIELD ISOTROPIC CONCRETE is used to define the heat of hydration for different concrete types. This depends on the chemical composition of the cement. Cement types I, II, III and V are catered for. Note that cement type IV is no longer widely used as admixtures tend to be used instead, therefore no data is available for this type. Provision has been made for a user defined chemical composition which is defined as cement type VI.

```
MATERIAL PROPERTIES FIELD ISOTROPIC CONCRETE  n
imat  k  T  C  H
icem  wcem  wcra  wslg  wpfa  pfacao  Tr  conv  itime
[C3S  C2S  C3A  C4AF  FreeCaO  SO3  MgO  Blaine]
```

imat The material property identification number
n Phase change type:
=0 Phase change not required (default)
=1 Del Giudice
=2 Lemmon
nset CARTESIAN SET number used to define local axes direction. If nset=0, the local axes coincide with global axes.
kx, ky, kz Thermal conductivity in x, y and z directions
T Reference temperature.
C Specific heat coefficient (only required for transient analysis).
H Enthalpy (only required for phase change analysis).
icem Cement type (1= Type I, 2= Type2 II, 3= Type III, 5=Type V, 6=user defined)
wcem Weight of cement per unit volume (must be defined in kg/m^3)
wcra Water/cementitious ratio (weight of water per unit volume/(wcem+wslg+wpfa))
wslg Weight of GGBF slag per unit volume (must be defined in kg/m^3)
wpfa Weight of fly ash per unit volume (must be defined in kg/m^3)

pfacao	CaO content of fly ash (defined as %) Values usually lie in the range 2-30%, class C fly ash = 24% and class F = 11%
Tr	Temperature at which concrete is assumed to cure (degrees C)
conv	Factor to convert heat (computed in J/m^3) to model unit system
itime	Time unit identifier (1=hrs,2=days)
C3S	User defined C_3S content (defined as %)
C2S	User defined C_2S content (defined as %)
C3A	User defined C_3A content (defined as %)
C4AF	User defined C_4AF content (defined as %)
FreeCaO	User defined Free CaO content (defined as %)
SO3	User defined SO_3 content (defined as %)
MgO	User defined MgO content (defined as %)
Blaine	User defined Blaine index (must be defined in m^2/kg)

Notes

1. The time step and any termination response times must be specified in hours or days. If the timescale is in days; the thermal conductivity and heat transfer coefficients should be defined wrt days, i.e. J/day/m.C and J/day/m^2.C.
2. The parameters C3S, C2S, C3A, C4AF, FreeCaO, SO3, MgO and Blaine are only required for the user defined cement type, that is, icem=6.
3. The heat of hydration properties in the data file must be specified in kg/m/^{o}C units. The model itself can be in any units and Modeller will tabulate the appropriate conversion factor to convert J/m^3 to the User's model units. In other words, the User defines their data in consistent units for their model, Modeller transforms these to kg/m/^{o}C on tabulation and also tabulates the conversion factor.
4. The CaO content for fly ash must be specified if fly ash is to be included. Some typical values are Class C fly ash (24%) and Class F fly ash (11%).
5. Other typical thermal properties for concrete:
   Specific heat capacity = 1020 J/kg.C (mature concrete).
   Conductivity = 2880-4680 J/hr/m.C
   Convective heat transfer coefficient = 3600-72000 J/hr/m^2.C.

Orthotropic Concrete Heat of Hydration Model

The data section MATERIAL PROPERTIES FIELD ORTHOTROPIC CONCRETE is used to define the material properties for the plane orthotropic concrete/resin field model. The model is valid for 2D plane field problems.

MATERIAL PROPERTIES FIELD ORTHOTROPIC CONCRETE n

imat k_x k_y θ T C H

icem wcem wcra wslg wpfa pfacao Tr conv itime

```
[C3S  C2S  C3A  C4AF  FreeCaO  SO3  MgO  Blaine]
```

All terms are as defined for the Isotropic Concrete Heat of Hydration Model.

Orthotropic Solid Concrete Heat of Hydration Model

The data section MATERIAL PROPERTIES FIELD ORTHOTROPIC SOLID CONCRETE is used to define the material properties for orthotropic solid concrete field problems. The model is valid for 3D solid field problems.

```
MATERIAL PROPERTIES FIELD ORTHOTROPIC SOLID  CONCRETE
n
imat  nset  kx  ky  kz  T  C  H
icem  wcem  wcra  wslg  wpfa  pfacao  Tr  conv  itime
[C3S  C2S  C3A  C4AF  FreeCaO  SO3  MgO  Blaine]
```

All terms are as defined for the Isotropic Concrete Heat of Hydration Model.

Linear Variation Convection/Radiation Model

Field link elements permit the variation of conductive, convective and radiative heat transfer to be related to an initial value at full closure, and a linear change in value with increasing gap opening.

The data section MATERIAL PROPERTIES FIELD LINK 18 is used to define the material properties for the linear variation convection/radiation model.

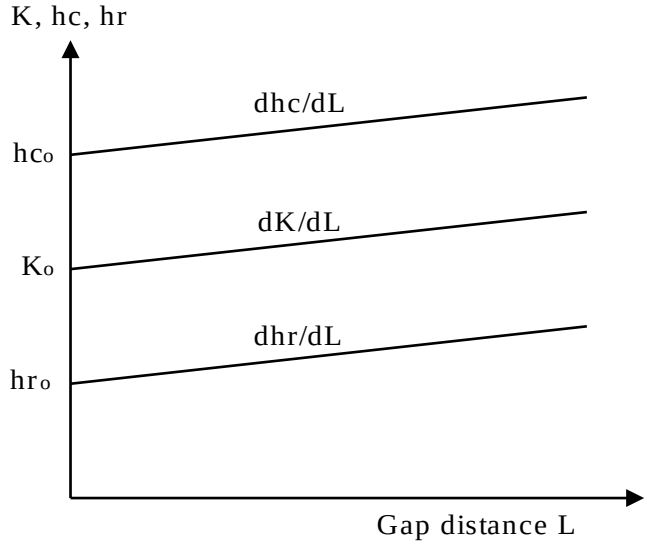
```
MATERIAL PROPERTIES FIELD LINK 18
imat  ko  [hco hro dk/dl dhc/dl dhr/dl T]
```

imat	The material property identification number
Ko	Gap conductance at origin
hco	Convective heat transfer coefficient at origin
hro	Radiative heat transfer coefficient at origin
dk/dL	Variation of gap conductance with opening distance
dhc/dL	Variation of gap convective heat transfer coefficient with opening distance
dhr/dL	Variation of radiative heat transfer coefficient with opening distance
T	Reference temperature.

Notes

1. If a negative value of a material property is calculated, then the material property is set to zero.
2. The figure below shows heat transfer coefficients/gap distance relationship for the linear variation convection/radiation model.

3. When a radiative heat transfer coefficient is specified the temperature units for the problem will be Kelvin by default. Option 242 allows temperatures to be input and output in Celsius (Centigrade) for problems involving radiative heat transfer.



Heat Transfer Coefficients/Gap Distance Relationship for the Linear Variation Convection/Radiation Model

Nonlinear Variation Convection/Radiation Model

The nonlinear convection/radiation field model permits the variation of conductive, convective and radiative heat transfer to be related to the gap opening distance in an arbitrary manner, by defining the variation of property with distance as a sequential series of straight line segments.

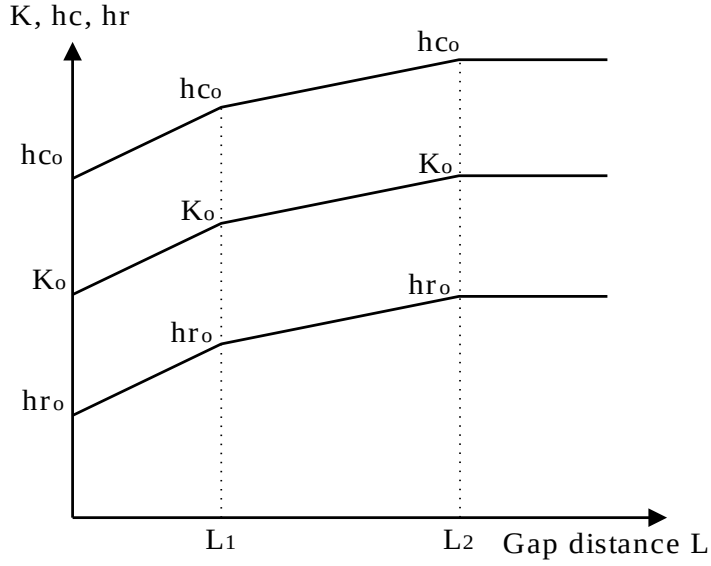
The data section MATERIAL PROPERTIES FIELD LINK 19 is used to define the material properties for the nonlinear variation convection/radiation model.

```
MATERIAL PROPERTIES LINK 19  N
imat  < ki hci hri Li >i=1,N  T
```

imat	The material property identification number
N	The number of points used to define the material properties
ki	Gap conductance for point **i**
hci	Convective heat transfer coefficient for point **i**
hri	Radiative heat transfer coefficient for point **i**
Li	Total distance for point **i** from origin
T	Reference temperature.

Notes

1. The figure below shows the heat transfer coefficients/gap distance relationship for the nonlinear variation convection/radiation model.
2. When a radiative heat transfer coefficient is specified the temperature units for the problem will be Kelvin by default. Option 242 allows temperatures to be input and output in Celsius (Centigrade) for problems involving radiative heat transfer.



Heat Transfer Coefficients/Gap Distance Relationship for the Nonlinear Variation Convection/Radiation Model

Material Assignments

Material specified using MATERIAL PROPERTIES, PLASTIC DEFINITION, VISCOUS DEFINITION, DAMAGE PROPERTIES and SHRIHKAGE is assigned to elements using the MATERIAL ASSIGNMENTS data chapter.

```
MATERIAL ASSIGNMENTS  [TITLE  title]

{L Llast Ldiff | G  igroup}  imat  [nset ipls icrp idam
imatv ivse iptm ishr]
```

L Llast Ldiff First, last and difference between elements with the same element group number

G Command word which must be typed to use element groups (see Group).

igroup The element group reference number.

imat The material identification number.

nset The Cartesian set number defining orthotropy of material properties (see Cartesian Sets and *Notes*).
ipls The stress potential set identification number. See Plastic Definition
icrp The creep property identification number. See Viscous Definition
idam The damage property identification number.
imatv The varying material identification number if it is defined in the pre-processing LUSAS Modeller model. This number is saved in the LUSAS Modeller results file for post-processing.
ivse The viscoelastic property identification number. See Viscous Definition
iptm The two-phase material identification number.
ishr The shrinkage property identification number.

Notes

1. **nset** is only applicable to material models using **nset** in their material input parameters. When **nset** is included in the MATERIAL ASSIGNMENT command the value input using MATERIAL PROPERTIES is overridden.
2. If an element is repeated the new properties/rigidities overwrite the previous values for that element.
3. Material properties may only be modified for a transient, dynamic or nonlinear problems subject to the following conditions:
    - Only the material assignment respecified will be modified
    - Further assignments may only use material properties datasets defined at the beginning of the problem.

Composite Material

Composite material input may be used to laminate a variety of materials together within a single element. In this lamination procedure, the composite lay-up sequence is always defined sequentially from the lower to upper surfaces of the element. Any appropriate LUSAS material model (see *Notes*) may be assigned to any defined layer within the element; hence combinations of material assignments may be used within a single element to achieve a numerical model of the laminated or composite material.

The composite material data is input in three stages.

1. The constitutive materials are defined using MATERIAL PROPERTIES, PLASTIC DEFINITION, VISCOUS DEFINITION and/or DAMAGE PROPERTIES.
2. The composite lay-up is defined using the COMPOSITE MATERIAL command.
3. The defined composite material property sets are assigned to elements using COMPOSITE ASSIGNMENTS.

The data section COMPOSITE MATERIAL is used to define the lay-up sequence for each composite property set. Data is specified in tabular form with each row relating to a particular layer in the sequence. The columns contain property set numbers relating to

previously defined MATERIAL PROPERTIES, STRESS POTENTIAL, CREEP PROPERTIES, DAMAGE PROPERTIES, VISCO ELASTIC PROPERTIES and SHRINKAGE. The STRESS POTENTIAL property set must have been previously defined under either the PLASTIC DEFINITION or VISCOUS DEFINITION data chapters.

COMPOSITE MATERIAL [TITLE title]

TABLE icmp

$imat_i$ [$angle_i$ $ipls_i$ $icrp_i$ $idam_i$ $lname_i$ $ivse_i$ $ishr_i$]

icmp	Composite material set number
nlayr	Number of layers in the composite property set (**i=1 to nlayr**)
imat$_i$	Material property set number for layer **i**
angle$_i$	Material direction angle for layer **i** (see *Notes*)
ipls$_i$	Stress potential set number for layer **i**
icrp$_i$	Creep property set number for layer **i**
idam$_i$	Damage property set number for layer **i**
iname$_i$	Layer name for layer
ivse$_i$	Viscoelastic property set number for layer **i**
ishr	Shrinkage property set number for layer **i**

Notes

1. **angle** is applicable to orthotropic material models. The value for angle defined using the COMPOSITE MATERIAL command overrides the value input with MATERIAL PROPERTIES.
2. The number of layers defined in a COMPOSITE MATERIAL table must equal the number of layers defined in the corresponding COMPOSITE GEOMETRY table when assigning values to elements.
3. A composite property set may consist of up to 99 layers.
4. Layer stresses are output by requesting output at Gauss points. All layers will then be output.
5. Composite material data and geometry is allocated to elements using the COMPOSITE ASSIGNMENTS command.
6. The composite lay-up defined within this data chapter is orientated with respect to either the local element axes or with respect to a CARTESIAN SET defined within the COMPOSITE ASSIGNMENTS data chapter. Note that an **angle** of 0° aligns with the appropriate x axis and an **angle** of 90° with the y axis.
7. For shell elements an appropriate plane stress material model must be used while for solid elements a 3D continuum model should be used (see the *LUSAS Element Reference Manual*).

Composite Assignments

Composite material and geometry sets are assigned to elements using the COMPOSITE ASSIGNMENTS data chapter.

COMPOSITE ASSIGNMENTS [TITLE title]

{L L_{last} L_{diff} | G igroup} icpm icpg [ns_1 $ns_{j,\ j=2,nnode}$]

L Llast Ldiff First, last and difference between elements with the same element group number.

G Command word which must be typed to use element groups (see Group).

igroup The element group reference number.

icpm The composite material identification number (see Composite Material).

icpg The composite geometry identification number (see Composite Geometry).

nsj The orientation of the composite lay-up at node **j** (see *Notes*).
= 0 for composite orientation with respect to local element axes (see *LUSAS Element Reference Manual f*or element axes definitions)
= h for composite orientation with respect to a defined CARTESIAN SET (see Cartesian Sets).

nnode Number of nodes

Notes

1. The number of layers defined by the COMPOSITE MATERIAL set and the COMPOSITE GEOMETRY set must be identical when used in the same assignment.
2. The composite lay-up orientation can be defined at each node where the node ordering is defined by the element topology. If the composite orientation is the same for all nodes then only the composite orientation for node 1 need be specified.
3. If the orientations of the composite lay-up are omitted then orientation with respect to the element local axes is assumed (**nsj** = 0 for **j**=1,**nnode**).
4. Orientations with respect to local element axes cannot be mixed with orientations with respect to CARTESIAN SETS for a particular element: either all nodes must have orientations with respect to the local element axes or with respect to CARTESIAN SETS (although the CARTESIAN SET numbers may be different for each of the nodes).

Element Ages

The element ages data chapter is used to define the time in days between casting and activation when using the CEB-FIP material model.

ELEMENT AGES

L L_{last} L_{diff} age

L Llast Ldiff	First, last and difference between elements numbers
age	Element age in days (default=0 days, see Notes)

Note

The element age should be defined as the time in days between casting and the time at which an element is activated. For example, an element could be 40 days old but activated in the analysis on day 28.

Activate/Deactivate Elements

The activation and deactivation facility accounts for the addition or removal of parts of a model as required by the simulation process. Also known as birth and death, rather than add or remove elements, the facility activates and deactivates elements to model their presence and absence. Staged construction processes, such as tunnelling, are an example of its use with structural analyses.

All elements to be used in the model are specified at the start of the analysis. To model the absence of a part of the model, elements defining it are deactivated. In structural analyses, these elements have their stiffness matrix reduced in magnitude, while for field analysis the conductivity matrix (or other analogous quantity) is reduced. This ensures the deactivated elements have a negligible effect on the behaviour of the remaining model. The element stresses and strains, fluxes and gradients and other analogous quantities are all set to zero.

To model the addition of a part to the model, the elements defining it are *activated*. In structural analyses, an unmodified stiffness matrix is computed for these elements and the activated elements are introduced in a stress/strain free state, except for any initial stresses or strains that have been defined. Strains are incremented from the point of activation and the current geometry is used to define the activated element's initial geometry. In field analyses activation works in the same manner, except that the quantities affected are the conductivity matrix (or other analogous quantity), the fluxes and the gradients.

By default, all loads applied to deactivated elements are initialised to zero and concentrated loads at nodes common to both active and inactive elements are shuffled to the active element. By setting option 385, however, loads applied to deactivated elements are preserved to enable reapplication if and when the elements are re-activated.

DEACTIVATE ELEMENTS

L L_{last} L_{diff} [ninc rdfact]

L Llast Ldiff	The first element, last element and difference between elements of the series of elements to be deactivated.

ninc The number of increments over which the fraction of residual force is to be redistributed, see *Notes* (default=1).

rdfact The fraction of residual force to be redistributed, see *Notes* (default=0.0).

ACTIVATE ELEMENTS

L L_{last} L_{diff} [ninc]

L Llast Ldiff The first element, last element and difference between elements of the series of elements to be activated.

ninc The number of increments over which any remaining residual force is to be redistributed, see *Notes* (default=1).

Notes

1. Restrictions on use. Elements cannot be activated or deactivated in the following circumstances:
   - In explicit dynamics analyses.
   - In Fourier analyses.
   - When using updated Lagrangian or Eulerian geometric nonlinearity.
   - Elements adjacent to slideline surfaces cannot be activated or deactivated.
2. Option 272 and NONLINEAR CONTROL must be specified when elements are to be activated or deactivated.
3. For **rdfact**=0.0, all internal forces associated with deactivated elements will remain in the system (i.e. the stresses, displacements, etc., for the remaining elements in a static structural analysis will remain unchanged if the external load remains constant. The same applies to temperature, fluxes, etc., in a static field analysis)'

   For **rdfact**=1.0, all internal forces associated with deactivated elements are removed from the system (i.e. the stresses, displacements, etc., for the remaining elements in a structural analysis will change. The same applies to temperature, fluxes, etc., in a field analysis)

   For 1.0>**rdfact**>0.0, a fraction of the internal forces in deactivated elements is removed with the remainder retained in the system for subsequent redistribution (e.g. in a structural analysis with **rdfact**=0.1, 10% of internal force is removed so that some stress redistribution takes place on the deactivation stage, 90% is retained for redistribution when the element is subsequently re-activated).
4. When deactivated elements are re-activated, any remaining internal forces associated with the re-activated elements (forces retained from deactivation stage) will be removed from the system so that a stress/flux redistribution takes place.
5. Deactivation and activation can take place over several increments if convergence difficulties are encountered by specifying the parameter **ninc**. For example, if **ninc**=3 and **rdfact**=0.3, then 10% of the internal force will be removed on the first increment, 20% on the second and 30% on the third. The TERMINATION section in NONLINEAR CONTROL (or alternative CONTROL chapter if

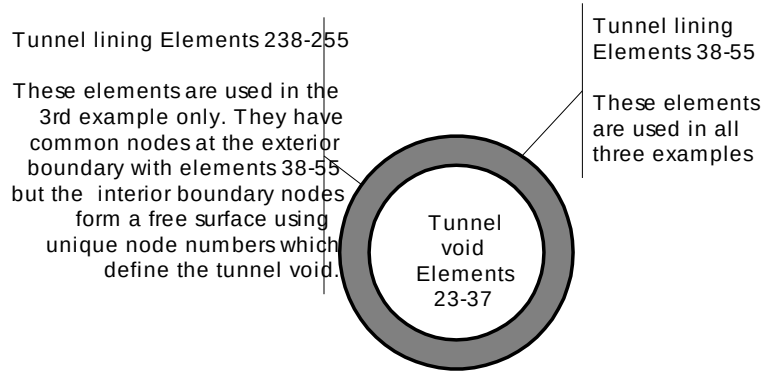
applicable) would then have to be defined to cover at least 3 increments (time steps). See the example on Data Input Examples for Tunnel Excavation.

6. Deactivated elements remain in the solution but with a scaled down stiffness/conductivity/etc., so that they have little effect on the residual structure. The scaling is performed using SYSTEM parameter **`STFSCL`** (default=1E-6) which can be changed by the user. In dynamic analyses, the mass and damping matrices are initialised to zero.
7. When an element is deactivated, by default all loads associated with that element are removed from the system and will not be reapplied if an element is subsequently activated. This includes concentrated loads, unless the load is applied to a boundary between active and inactive elements when the load is shuffled to the active element. The only exceptions are prescribed loads (displacement, temperature or other analogous quantity) which can be applied to inactive elements. Accelerations may also be applied in a dynamic analysis but this is not recommended.

   Setting option 385 overrides the default load initialisation, preserving loads applied to deactivated elements to enable their reapplication if and when the elements are re-activated.
8. When an element is activated in a subsequent stage of a structural analysis, it is introduced in an unstressed/unstrained state and the initial element geometry is taken as the current geometry. Strains are incremented from the time at which the element is activated. Initial stresses/strains and residual stresses may be defined for an element at the re-activation stage. The same applies to field analyses with analogous quantities of flux and gradient, although a geometry update does not take place.
9. The activation of an element that is currently active results in the initialisation of stresses, strains, fluxes and gradients along with an update of the initial geometry to the current geometry. The element is considered to have just become active. The internal equilibrating forces that currently exist in the element will be immediately redistributed throughout the mesh. This provides a simplified approach to redistribute all the element forces of this newly activated element in one increment, for cases in which no relaxation of the remaining structure was required. See example on Data Input Examples for Tunnel Excavation.
10. The direction of local element axes can change during an analysis when elements are deactivated and reactivated. In particular, 3-noded beam elements that use the central node to define the local axes should be avoided as this can lead to confusion. For such elements the sign convention for bending moments for a particular element may change after re-activation (e.g. it is recommended that BSL4 should be preferred to BSL3 so that the 4th node is used to define the local axes and not the initial element curvature).
11. Care should be taken when deactivating elements in a geometrically nonlinear analysis, especially if large displacements are present. It may be necessary to apply prescribed displacements to deactivated elements in order to attain a required configuration for reactivation.

12. It should be noted that the internal forces in the elements will not balance the applied loading until all residual forces in activated/deactivated elements have been redistributed.

Data Input Examples for Tunnel Excavation


Tunnel lining Elements 238-255
These elements are used in the 3rd example only. They have common nodes at the exterior boundary with elements 38-55 but the interior boundary nodes form a free surface using unique node numbers which define the tunnel void.
Tunnel lining Elements 38-55
These elements are used in all three examples
Tunnel void Elements 23-37

Element Numbers Defining the Tunnel Lining and Void

Example 1. Residual Force Redistribution in Increments During Deactivation

```
      .
      .
```

```
MATERIAL PROPERTIES
C concrete properties
1   14E9  0.3  2400
C soil properties
2   15E7  0.3  2000
MATERIAL ASSIGNMENTS
C assign all elements with soil material properties
1 200 1  2
SUPP NODE
1 59 1  R R
C Apply load to set up initial stress state in soil and
maintain constant (however load
C does not have to remain constant)
```

```
LOAD CASE
C surface surcharge
CL
60 95 1  0  -5000
C soil self weight
CBF
1 200 1  0  -2000
NONLINEAR CONTROL
ITERATION 10
CONVERGENCE  0.1 0 0.1 0 0 0
OUTPUT 0 1 1
C Deactivate the elements representing the excavated
material.
DEACTIVATE ELEMENTS
C  f   l   i  number of increments
redistribution factor
C         (default=1)       (default=0.0)
   23  55  1        3                        0.5
C This will have the effect of deactivating elements 23
to 55 and redistributing 0.5 of
C the equilibrating forces to the remaining mesh prior
to the activation of elements. This
C can be interpreted as a relaxation of the stresses
around the tunnel excavation, i.e
C displacements and stresses will change so that the
boundary of the excavation changes
C shape. If factor=0.0 then all the residual force will
be stored for subsequent
C redistribution. This redistribution stage will take
place over the next 3 increments.
C i.e.  Residual force =    (1 - 0.5*1/3)Fr   inc 2
C              (1 - 0.5*2/3)Fr       3
C              (1 - 0.5)Fr     4
C More than one increment will only be required if
convergence difficulties occur. For
C linear elastic materials the same final result will be
obtained irrespective of the
```

```
C number of increments used to redistribute the
residual. For nonlinear materials it may
C not be possible to redistribute the force in one
increment. Allowing a staged
C redistribution gives the opportunity to change the
applied loading during this process.
C Redistribution increments (initial loading is
preserved except for load on deactivated
C elements):
NONLINEAR CONTROL
TERMINATION  0  3
MATERIAL ASSIGNMENTS
C Redefine material properties for tunnel lining
38 55 1   1
C Reactivate elements modelling the tunnel lining
ACTIVATE ELEMENTS
C  f   l   i    number of increments (default=1)
   38  55  1          4
C This command will re-activate elements 38 to 55 in an
unstressed/unstrained state and
C any remaining residual equilibrating force associated
with these elements will be
C redistributed throughout the mesh over the next 4
increments.
C i.e.    Residual force =  0.75*Fr'    inc 5
C               0.5*Fr'         6
C               0.25*Fr'        7
C               0.0*Fr'         8
C where Fr'= (1-0.5)Fr (from above)
C Redistribute the remaining residual forces
NONLINEAR CONTROL
TERMINATION  0  4
END
```

Example 2. Simplified Version with No Redistribution at the Deactivation Stage

```
    .
    .
MATERIAL PROPERTIES
C concrete properties
1     14E9   0.3  2400
C soil properties
2     15E7   0.3  2000
MATERIAL ASSIGNMENTS
C assign all elements with soil material properties
1 200 1   2
SUPP NODE
1 59 1  R R
C Apply load to set up initial stress state in soil
LOAD CASE
C surface surcharge
CL
60 95 1   0   -5000
C soil self weight
CBF
1 200 1  0   -2000
NONLINEAR CONTROL
ITERATION 10
CONVERGENCE  0.1 0 0.1 0 0 0
OUTPUT 0 1 1
MATERIAL ASSIGNMENTS
C Redefine material properties for tunnel lining
38 55 1   1
C Activate elements modelling the tunnel lining
ACTIVATE ELEMENTS
C  f   l   i    number of increments (default=1)
   38  55  1          1
C This command will activate elements 38 to 55 and
restore them to an unstressed/
```

```
C unstrained state while the internal equilibrating
force associated with these elements
C will be redistributed throughout the mesh. If it is
not required to allow some
C relaxation of the boundary of the excavation prior to
installing the tunnel lining, then
C the activation of the tunnel lining can be done in one
step. The activation of an
C element which is currently active results in a
resetting of the stresses/strains to zero
C and the element is considered to have just become
active.
C Deactivate the elements that represent the tunnel void
(i.e. the area inside the lining)
DEACTIVATE ELEMENTS
C  f   l   i   number of increments
redistribution factor
C         (default=1)       (default=0.0)
   23  37  1         1                        1.0
C This will have the effect of deactivating elements 23
to 37 which form the tunnel void
C and redistributing all residual force associated with
these elements
END
```

Example 3. Alternative Method of Defining the Tunnel Lining

```
    .
    .
MATERIAL PROPERTIES
C concrete properties
1     14E9   0.3 2400
C soil properties
2     15E7   0.3  2000
C At the outset, deactivate the elements that will
represent the tunnel lining on re-
C activation.
DEACTIVATE ELEMENTS
```

```
C  f   l   i  number of increments        redistribution
factor
C         (default=1)       (default=0.0)
   238 255 1        1                          1.0
C For simplicity, these elements may be thought of as
being superimposed on elements 38 to
C 55 which formed the tunnel lining in the previous
analyses. However, only the external
C boundary nodes of the tunnel lining must be common
with nodes in theunderlying mesh.
C In other words, these elements are connected to the
elements which represent the soil at
C the external boundary but the internal element
boundaries are discretised with different
C node numbers to form a free surface i.e. the tunnel
void.
MATERIAL ASSIGNMENTS
C assign soil material properties
1   200 1   2
C assign lining elements with concrete material
properties
238 255 1   1
SUPP NODE
1 59 1  R R
C fix the additional nodes defining the internal
boundary of the tunnel lining (i.e.
C located on the elements that are deactivated from the
outset). This permits the internal
C dimensions of the tunnel opening to be preserved if
critical. Note that prescribed
C displacements could also be applied to these nodes to
dictate the shape of the internal
C surface prior to activating the tunnel lining.
201 289 1  R R
C Apply load to set up initial stress state in soil.
LOAD CASE
C surface surcharge
```

```
CL
60 95 1 0   -5000
C soil self weight
CBF
1 200 1 0   -2000
NONLINEAR CONTROL
ITERATION 10
CONVERGENCE  0.1 0 0.1 0 0 0
OUTPUT 0 1 1
C Activate elements modelling the tunnel lining
ACTIVATE ELEMENTS
C  f   l   i    number of increments (default=1)
   238 255 1          1
C Deactivate the elements that represent the overall
tunnel excavation
DEACTIVATE ELEMENTS
C  f   l   i   number of increments
redistribution factor
C         (default=1)       (default=0.0)
   23  55  1         1                        1.0
SUPP NODE
C Release the nodes modelling the interior walls of the
lining
201 289 1  F F
END
```

Damping Properties

The data section DAMPING PROPERTIES is used to define the frequency dependent Rayleigh damping parameters for elements which contribute to the damping of the whole structure. This section is valid for viscous (modal) and structural (hysteretic) damping and may be utilised when distributed viscous and/or structural damping factors are required using MODAL_DAMPING CONTROL (see Modal Damping Control).

DAMPING PROPERTIES [VISCOUS | STRUCTURAL]

L L_{last} L_{diff} **<** $(a_r\ b_r\ \omega)_i$ **>**$_{i=1,n}$

L Llast Ldiff First, last and difference between elements with identical damping properties.

ar Mass Rayleigh damping parameter.

br Stiffness Rayleigh damping parameter.

ω Circular frequency at which the damping parameters apply.

n Number of frequencies for which Rayleigh parameters are specified. If more than one set of parameters are specified linearly interpolated parameters will be computed at the required frequency.

Notes

1. If only **a** and **b** are input it is assumed the Rayleigh parameters are the same for all frequencies.
2. The Rayleigh parameters are interpolated at the required frequency.
3. If this data chapter is omitted and distributed damping factors are required, then Rayleigh parameters from material properties are used.

Slidelines

Slidelines may be used to model contact and impact problems, or to tie dissimilar meshes together. Several slideline options are available:

- **Tied sliding**
- **General sliding without friction**
- **General sliding with friction**
- **Sliding only (without friction or lift off)**
- **Null**

The tied slideline option allows meshes of differing degrees of refinement to be connected without the need of a transition zone between the meshes. It can be very useful in creating a highly localised mesh in the region of high stress gradients.

The friction/no-friction slideline types model the finite relative deformation of contacting bodies in two or three dimensions where the contact is stationary or sliding, constant or intermittent. The sliding only option is similar but does not permit intermittent contact, i.e. the surfaces are kept in contact, allowing frictionless sliding contact without lift-off to be modelled. A null slideline is ignored during the analysis, useful when performing a preliminary check on a model.

Each slideline comprises two surfaces - the master surface and the slave surface. These surfaces are specified using nodes in the region where contact is expected to occur. The nodes are ordered to form contact segments where each segment is a boundary face of an underlying element.

Note that, except for tied slidelines, the slideline contact facility is inherently nonlinear and must be used in a nonlinear analysis. This requires the use of the NONLINEAR CONTROL data chapter.

The slideline facility requires the following data input:

- Slideline property definition
- Slideline surface definition
- Slideline assignments

The SLIDELINE PROPERTIES command specifies the properties of each slideline, such as the stiffness scale factor and the coefficient of friction. The topology of each slideline surface is specified using the SLIDELINE_SURFACE DEFINITION data section. The segment ordering defined in this section should be checked before a full analysis is undertaken. Alternatively, or in conjunction with SLIDELINE_SURFACE DEFINITION, the AUTOMATIC SURFACE command can be used to define all valid external surfaces as possible contact surfaces. The slideline surfaces are associated with the required slideline properties using the SLIDELINE ASSIGNMENT data section. With automatically generated contact surfaces, The SLIDELINE ASSIGNMENT MATERIAL data section must be used.

The slideline type, (e.g. tied, general sliding with friction), can be redefined during an analysis. This involves re-specifying the SLIDELINE ASSIGNMENT data chapter. SLIDELINE PROPERTIES can also be redefined and assigned.

Temperature dependent SLIDELINE PROPERTIES can also be specified. In this case the TABLE command must follow the SLIDELINE PROPERTIES command. Lines of data listing the slideline properties at particular reference temperatures are then input. The stiffness scale factors and the coefficient of friction are linearly interpolated across the reference temperatures. All other properties remain unchanged.

A nonlinear friction law can be introduced by using the SLIDELINE PROPERTIES USER command. This command allows a set of friction parameters to be defined that can vary with the temperature, velocity and acceleration of the contacting surfaces. These properties may also be specified as temperature dependent.

The following features are available with slidelines

- **Contact constraint enforcement** By default the penalty method is used. The augmented Lagrangian method is also available.
- **Geometric definition** Slideline surfaces can be modelled using linear/bi-linear segments, or as curved contact surfaces using quadratic patches. With quadratic patches the curved contact geometry is constructed from a patch of slideline segments, while the contact forces are distributed to the closest segment. The quadratic patches and the curved geometry are set-up automatically within LUSAS Solver. The standard patch consists of two linear segments in 2D and four bi-linear segments (quadrilateral or triangular) in 3D.

Where a patch definition is not possible the standard linear/bi-linear definition is used instead.

- **Rigid surfaces** Rigid slideline surfaces are available for modelling contact with rigid bodies. Rigid surfaces can be assigned to valid structural elements as well as to special rigid surface elements R2D2, R3D3 and R3D4. The latter are recommended for modelling rigid bodies since they remove the need for defining structural elements, hence speeding up the solution. All nodes on a rigid surface should be completely restrained. Rigid surfaces cannot contact each other so only one slideline surface can be defined as rigid – master or slave.
- **Close contact** This defines a region above a slideline surface within which a soft spring is applied, but with no force. The stiffness of this spring is applied to all nodes that lie within the close-contact region, thus softening the transition between in-contact and out-of-contact states. This can help the nonlinear convergence when in-contact/out-of-contact chatter is experienced.
- **Contact cushioning** This facility applies a contact force and stiffness above a surface that increases exponentially as a node moves closer to the surface. This cushions the impact of a node with the surface and softens the transition between in-contact and out-of-contact states. This can help stabilise in-contact/out-of-contact chatter and any consequent nonlinear convergence difficulties.
- **Slideline extensions** The boundary of a slideline segment can be expanded by specifying a slideline extension. Points outside the segment but within the extended boundary are considered valid for contact. This is particularly useful near the edges of a slideline surface, where a node could be on a segment in one nonlinear iteration and off the segment in the next iteration – a form of chatter that can cause nonlinear convergence difficulties.
- **Pre-contact** This facility brings two bodies into initial contact using interface forces that act between the slideline surfaces. One of the surfaces must be free to move as a rigid body and the direction of movement is dictated by the interface forces (which act normal to the surface), applied loading and support conditions.

  Pre-contact can be used to overcome problems encountered when applying an initial load (other than prescribed displacement) to a discrete body that, without the slideline, would undergo unrestrained rigid body motion. This is particularly the case when an initial gap exists between the contacting surfaces and a load is applied to bring them into contact. Pre-contact is only applicable to static analyses.

  Warning. Incorrect use of this procedure could lead to initial straining in the bodies or to an undesired starting configuration.

The following table gives a list of elements valid for use with slidelines.

Element type	LUSAS elements
Engineering beams	BMS3, BTS3
Thick shells	TTS3, QTS4
Plane stress continuum	TPM3, TPM3E, TPK6, TPM6, QPM4, QPM4E, QPM4M, QPK8, QPM8
Plane strain continuum	TNK6, TPN3, TPN3E, TPN6, QNK8, QPN4, QPN4E, QPN4L, QPN4M, QPN8
Axisymmetric solid continuum	TAX3, TAX3E, TAX6, TXK6, QAX4, QAX4E, QAX4L, QAX4M, QAX8, QXK8,
Solid continuum	TH4, TH4E, TH10, TH10K, PN6, PN6E, PN6L, PN12, PN12L, PN15, PN15K, PN15L, HX8, HX8E, HX8L, HX8M, HX16, HX16L, HX20, HX20K, HX20L
Continuum two-phase	TH10P, TPN6P, PN12P, PN15P, HX16P, HX20P, QPN8P
2D interface	IAX4, IAX6, IPN4, IPN6
3D interface	IS6, IS8, IS16, IS12
2D rigid surface	R2D2
3D rigid surface	R3D3, R3D4

Slideline Properties

The data section SLIDELINE PROPERTIES is used to define the overall features of each slideline, such as the interface stiffness scale factor and friction coefficient. If a table of temperature dependent properties is specified under this data chapter, linear interpolation will be applied to all material data. It is possible to redefine and assign the SLIDELINE PROPERTIES as an analysis progresses.

SLIDELINE PROPERTIES [TITLE title]

isprop M_{scale} S_{scale} μ α_r extn T C_{cont} Pentol Pupfac T_z

isprop Slideline property assignment number

M_{scale} Interface stiffness scale factor for master surface (see *Notes*)
For automatic surfaces factor is applied to IMAT1, where IMAT1 is a material property identifier
Explicit solution schemes (default = 0.1)
Implicit/static solution schemes (default = 1.0)
Tied slidelines (default=100.0)

S_{scale} Interface stiffness scale factor for slave surface
For automatic surfaces factor is applied to IMAT2, where IMAT2 is a material property identifier

	Explicit solution schemes (default = 0.1) Implicit/static solution schemes (default = 1.0) Tied slidelines (default=100.0)
μ	Coulomb friction coefficient (default = 0.0) (see *Notes*)
α_r	Zonal contact detection parameter (default=0.01) (see *Notes*)
extn	Slideline extension distance (default = 0.0) The extension eliminates interpenetration for slideline surfaces which are significantly irregular. More details may be found in the *LUSAS Theory Manual*.
T	Reference temperature (default = 0.0)
C_{cont}	Close contact detection parameter, see *Notes* (default=0.001)
Pentol	Penetration tolerance with augmented Lagrangian method (see *Notes*)
Pupfac	Penalty update factor with augmented Lagrangian method (default=10.0) (see *Notes*)
T_z	Tied slideline detection zone (default a large number) (see *Notes*)

Notes

1. The stiffness scale factors control the amount of interpenetration between the two slideline surfaces. Increasing these factors will decrease the relative penetration.
2. The zonal contact detection parameter in conjunction with the system parameter BBOXF controls the extent of the contact detection test. In the first phase a bounding box is defined around each body or contact surface in turn permitting an efficient check for their overlap. BBOXF is a scaling factor applied to this box, the default value of 1.2 is sufficient for most analyses. For the first iteration of pre-contact analyses BBOXF is replaced by the system parameter BBOXFP whose default value is 10.0. Then overlap is checked between bounding boxes containing individual surface segments and a nodal volume around the contacting node. The zonal detection distance sets the size of the nodal volume for a node and is calculated by multiplying the zonal detection parameter and the largest dimension of the model in the coordinate directions. A value of 1 will force a search of every single possible contact segment. The default value of 0.01 searches a zone which is 1% of the model size centred on the contacting node.
3. The close contact detection parameter, **Ccont**, is used to check if a node is threatening to contact a slideline surface. The surface tolerance used is the product of the detection parameter and the length of the surface segment where the node is threatening to penetrate. If the distance between the surfaces is less than the surface tolerance a spring is included at that point just prior to contact. By default, the stiffness of the spring is taken as 1/1000 of the surface stiffness. The factor of 1/1000 can be changed by redefining the SYSTEM parameter **SLSTCC**. The inclusion of springs in this manner helps to stabilise the solution algorithm when surface nodes come in and out of contact during the iteration process. The close contact detection facility is not available for explicit dynamics.
4. Slideline properties can be redefined at selected stages in an analysis by re-specifying SLIDELINE PROPERTIES.

5. The explicit tied slideline option is more robust when the mesh with the greatest node density is designated the slave surface.
6. Scaling of the slideline surface stiffnesses is automatically invoked at the beginning of each analysis if the ratio of the average stiffness values for each constituent slideline surface differ by a factor greater than 100 (a default value, modifiable using the **SLSTFM** SYSTEM parameter). In this manner account is made for bodies having significantly different material properties. Option 185 will suppress this facility.
7. The Coulomb friction can only be specified for slideline type 2.
8. The augmented Lagrangian scheme aims to reduce penetrations to below the penetration tolerance **Pentol**. If 0.0 is specified for the penetration tolerance, a machine dependant near-zero value is used. On 32-bit Windows PCs, the default is $1x10^{-9}$.
9. If the penetrations are not reducing quickly enough in the augmented Lagrangian scheme, the penalty update procedure can be used to increase the penalty parameters (contact stiffnesses) in order to accelerate the reduction. The procedure identifies nodes where the reduction in penetration is slow and updates the penalty parameter in the following manner.

$$\varepsilon_N^{k+1} = \beta\varepsilon_N^k \qquad \text{if} \qquad \left(\frac{g_N^k}{g_N^{k-1}}\right) \geq \gamma$$

$$\varepsilon_N^{k+1} = \varepsilon_N^k \qquad \text{if} \qquad \left(\frac{g_N^k}{g_N^{k-1}}\right) < \gamma$$

   where ε_N is the penalty parameter (contact stiffness), g_N is the penetration, k is the augmented Lagrangian update number, β is **Pupfac** and γ=0.25.
10. The penetration tolerance **Pentol** and penalty update factor **Pupfac** are only valid with the augmented Lagrangian method
11. The tied slideline detection zone $\mathbf{T_z}$ is a dimensional value that sets the maximum distance from a node to the adjacent surface beyond which no contact element will be formed. For instance if a body is 1m long a value of 1mm will probably be acceptable. Contact elements will only be formed for nodes which are within 1mm of the surface.

User-Defined Slideline Properties

The data section SLIDELINE PROPERTIES USER allows user supplied subroutines defining a nonlinear friction law to be called from within LUSAS. This allows you to introduce a friction law which may depend on the velocity, acceleration or temperature of the contacting surfaces. The user subroutines can only be used to compute the friction parameters and the allowable tangential frictional force on a surface. The remaining input data is treated in the same way as data specified under SLIDELINE PROPERTIES. As temperatures are passed into the user routines, together with a table

of friction parameters, you can override interpolation of the friction parameters if necessary. It is possible to redefine SLIDELINE PROPERTIES USER as an analysis progresses.

Source code access is available to these interface routines and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. Contact FEA for full details of this facility. Since user specification of a nonlinear friction law involves the external development of source FORTRAN code, as well as access to LUSAS code, this facility is aimed at the advanced LUSAS user.

SLIDELINE PROPERTIES USER nfric [TITLE title]

isprop M_{scale} S_{scale} $< \mu_i >_{i=1,nfric}$ α_r **extn T** C_{cont} **Pentol Pupfac** T_z

nfric	Number of friction parameters
isprop	Slideline property assignment number
M_{scale}	Interface stiffness scale factor for master surface For automatic surfaces factor is applied to material IMAT1, where IMAT1 is a material property identifier Explicit solution schemes (default = 0.1) Implicit/static solution schemes (default = 1.0) Tied slidelines (default=100.0)
S_{scale}	Interface stiffness scale factor for slave surface For automatic surfaces factor is applied to material IMAT2, where IMAT2 is a material property identifier Explicit solution schemes (default = 0.1) Implicit/static solution schemes (default = 1.0) Tied slidelines (default=100.0)
μ_i	Friction parameters for user defined friction law
α_r	Zonal contact detection parameter (default = 0.01)
extn	Slideline extension distance (default = 0.0) The extension eliminates interpenetration for slideline surfaces which are significantly irregular. See *Notes* (more details may be found in the *LUSAS Theory Manual*).
T	Reference temperature (default = 0.0)
C_{cont}	Close contact detection parameter, see *Notes* (default=0.001)
Pentol	Penetration tolerance with augmented Lagrangian method (see *Notes*)
Pupfac	Penalty update factor with augmented Lagrangian method (default=10.0) (see *Notes*)
T_z	Tied slideline detection zone (default a large number) (see *Notes*)

Notes

1. The stiffness scale factors control the amount of interpenetration between the two slideline surfaces. Increasing these factors will decrease the relative penetration.

2. The zonal contact detection parameter in conjunction with the system parameter BBOXF control the extent of the contact detection test. In the first phase a bounding box is defined around each body or contact surface in turn permitting an efficient check for their overlap. BBOXF is a scaling factor applied to this box, the default value of 1.2 is sufficient for most analyses. For the first iteration of precontact analyses BBOXF is replaced by the system parameter BBOXFP whose default value is 10.0. Then overlap is checked between bounding boxes containing individual surface segments and a nodal volume around the contacting node. The zonal detection distance sets the size of the nodal volume for a node and is calculated by multiplying the zonal detection parameter and the largest dimension of the model in the coordinate directions. A value of 1 will force a search of every single possible contact segment. The default value of 0.01 searches a zone which is 1% of the model size centred on the contacting node.
3. The number of friction parameters defined must agree with the number expected **`nfric`**. All data values must be specified when using this data chapter, default values for data items other than the friction parameters will be set if **D** is specified in the required locations.
4. Slideline properties can be redefined at selected stages in an analysis by re-specifying SLIDELINE PROPERTIES USER.
5. The close contact detection parameter, **`Ccont`**, is used to check if a node is threatening to contact a slideline surface. The surface tolerance used is the product of the detection parameter and the length of the surface segment where the node is threatening to penetrate. If the distance between the surfaces is less than the surface tolerance a spring is included at that point just prior to contact. By default, the stiffness of the spring is taken as 1/1000 of the surface stiffness. The factor of 1/1000 can be changed by redefining the SYSTEM parameter **`SLSTCC`**. The inclusion of springs in this manner helps to stabilise the solution algorithm when surface nodes come in and out of contact during the iteration process. The close contact detection facility is not available for explicit dynamics.
6. The augmented Lagrangian scheme aims to reduce penetrations to below the penetration tolerance **`Pentol`**. If 0.0 is specified for the penetration tolerance, a machine dependant near-zero value is used. On 32-bit Windows PCs, the default is $1x10^{-9}$.
7. If the penetrations are not reducing quickly enough in the augmented Lagrangian scheme, the penalty update procedure can be used to increase the penalty parameters (contact stiffnesses) in order to accelerate the reduction. The procedure identifies nodes where the reduction in penetration is slow and updates the penalty parameter in the following manner.

$$\varepsilon_N^{k+1} = \beta\varepsilon_N^k \qquad \text{if} \qquad \left(\frac{g_N^k}{g_N^{k-1}}\right) \geq \gamma$$

$$\varepsilon_N^{k+1} = \varepsilon_N^k \qquad \text{if} \qquad \left(\frac{g_N^k}{g_N^{k-1}}\right) < \gamma$$

where ε_N is the penalty parameter (contact stiffness), g_N is the penetration, k is the augmented Lagrangian update number, β is **Pupfac** and γ=0.25.

8. The tied slideline detection zone $\mathbf{T_z}$ is a dimensional value that sets the maximum distance from a node to the adjacent surface beyond which no contact element will be formed. For instance if a body is 1m long a value of 1mm will probably be acceptable. Contact elements will only be formed for nodes which are within 1mm of the surface.
9. The penetration tolerance **Pentol** and penalty update factor **Pupfac** are only valid with the augmented Lagrangian method

Slideline Surface Definition

The data sections SLIDELINE_SURFACE DEFINITION is used to define the topology of each slideline surface.

SLIDELINE_SURFACE DEFINITION nsurf [TITLE title]

< N_i >$_{i=1,nseg}$

nsurf The surface number to be defined
N_i The node numbers defining each segment
nseg The number of nodes defining each segment

SLIDELINE_SURFACE DEFINITION RIGID nsurf [TITLE title]

< N_i >$_{i=1,nseg}$

nsurf The surface number to be defined
N_i The node numbers defining each segment
nseg The number of nodes defining each segment

Notes

1. A segment is defined as an element face which forms part of the possible contact surface. Two nodes are required to define the slideline for 2D analyses. Three or four nodes, are required to define a segment for a 3D problem, depending on the particular element face. The segments must be located along the object boundary with segment node ordering defined in a consistent direction, however, segments may be specified in an arbitrary order.
2. For 2D surfaces, the master surface segment node ordering must be defined such that outward normals are defined. The slave surface node ordering must then be in the same direction. If Option 61 is used, both the master and slave surface segment node ordering must be defined such that outward normals are defined. The examples below show these cases.
3. For 2D slidelines the positive local x axis for the master surface coincides with the direction of segment node ordering. A right hand screw rule is then applied with a

positive local z axis pointing out of the page. The positive local y axis defines the direction of the surface normal, see Example 1. 2D Slideline Surface Definition.

4. For 3D surfaces, the node definition for each surface segment must be labelled in an anti-clockwise direction (when looking towards the structure along the outward normal to the slideline surface). The numbering convention obeys the right hand screw rule. See the examples below.
5. Sharp corners (of approximately 90°) are generally best described by the use of two surfaces.
6. Coarse mesh discretisation in the region of contact should be avoided.
10. A slideline extension of the order of the anticipated allowable penetration is recommended for 3D slide surfaces.
7. The coordinates of all contact nodes which are determined to have penetrated prior to the commencement of the analysis are reset normally to the contacted surface. Option 186 will suppress this facility as required. This resetting of coordinates is not available for tied slidelines.
8. SLIDELINE_SURFACE DEFINITION RIGID is used to define a rigid surface. Such a surface is non-deformable and all nodes on it must be fully restrained. These restraints should be defined in the SUPPORT NODES data section.
9. Standard node-on-segment contact assumes an equal weighting of force and stiffness at each contact node. Specifying `isldst=1` uses a weighting based on contacted areas, which should improve the accuracy of the results.
10. Standard node-on-segment contact assumes an equal weighting of force and stiffness at each contact node. Specifying `isldst=1` uses a weighting based on contacted areas, which should improve the accuracy of the results.

Slideline Surface Automatic

The slideline contact surfaces can be defined automatically by including the command

SLIDELINE_SURFACE AUTOMATIC

Notes

1. All the external faces corresponding to valid underlying element types are extracted and united to form the contact surfaces. A valid element type is one that has edges or faces that can be used in a SLIDELINE SURFACE DEFINITION, however, elements with quadratic edges or faces are not currently supported.
2. The material identifiers, IMAT1 and IMAT2, belonging to the elements which have come into contact are used to identify the correct contact conditions set by the SLIDELINE ASSIGNMENTS MATERIAL command.
3. Care must be taken to avoid unintended contact when shells are present. A node is above the surface when it lies in the direction of the positive normal of the shell.
4. Manual slideline surfaces (defined by SLIDELINE_SURFACE DEFINITION) can be used in conjunction with the automatic surfaces. Nodes lying on a manually

defined surface can only come in contact with the corresponding master or slave surface. However, a node on an automatically generated surface can come in contact with a manually defined surface.

5. The external surface data is saved in a .ASL file. In the event of resolving the problem the nodal topology data is checked against the data stored in this file and if there are no changes the surface data is reused.
6. If the angle between two adjacent segments is less than or equal to the system parameter ANGLEC then it is deemed to be an edge. Slideline extensions can be applied at this point if they have been defined. ANGLEC has a default value of 120°.
7. The contact surfaces must be defined using linear geometry, i.e. curved quadratic patches (**islgem** = 2) are not currently supported.

Slideline Assignments

The data section SLIDELINE ASSIGNMENTS assembles manually defined slideline surfaces into the required slidelines.

SLIDELINE ASSIGNMENTS [TITLE title]

numsl M_{nsurf} S_{nsurf} isprop isltyp [P_{cont} islcsh islpss islgem isldst islaug islpup alstat alsymf]

numsl The slideline number

M_{nsurf} The master slideline surface number

S_{nsurf} The slave slideline surface number

isprop Slideline property number specified in SLIDELINE PROPERTIES data chapter

isltyp Type of slideline treatment required for this surface (see *Notes*) :-
- =0 for a null slideline
- =1 for general sliding without friction (default)
- =2 for general sliding with friction
- =3 for tied slidelines
- =4 for sliding only (without friction or lift off)

P_{cont} Pre-contact detection flag :-

no pre-contact process	= 0 (default)
pre-contact required	= 1 (see *Notes*)

islcsh Contact cushioning parameter

No contact cushioning	= 0 (default)
Contact cushioning required	= 1 (see *Notes*)

islpss Number of contact check passes (see *Notes*) :-

Two pass	= 2 (default)
One pass	= 1

islgem Contact surface geometric definition :-

Linear (2D), bi-linear (3D)	= 1 (default)
Curved quadratic patches	= 2 (see *Notes*)

isldst Use of distributed force/stiffness using contact areas (see *Notes*)
0 (off, default), 1 (on)

islaug Use of the augmented Lagrangian method (see *Notes*)
0 (off, default), 1 (on)

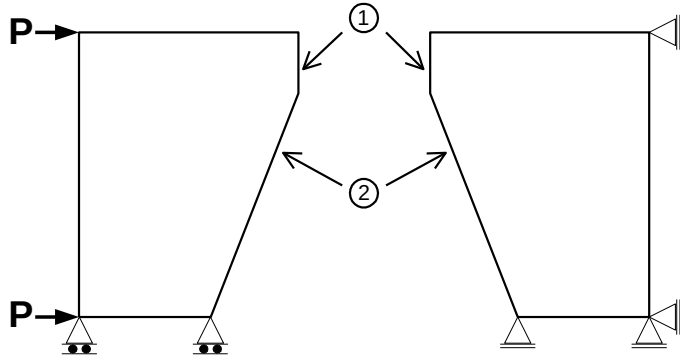
islpup Use of the penalty update procedure with the augmented Lagrangian method (see *Notes*)
0 (off, default), 1 (on)

alstat Contact status with the augmented Lagrangian method (see *Notes*)
Uzawa's method = 1 (default)
Powell's method = 2

alsymf Symmetrised Coulomb friction with the augmented Lagrangian method (see *Notes*)
0 (off, default), 1 (on)

Notes

1. The slideline surface numbers specified must be previously defined in the SLIDELINE_SURFACE DEFINITION command.
2. A slideline surface may only be used once.
3. Specifying a friction coefficient with **isltyp**=1 or 3 is invalid.
4. For 3D slideline surfaces the assignment of the master and slave surfaces is immaterial.
5. A null slideline may be specified if a slideline is defined in the data file but not required for a particular analysis.
6. Assignments can be changed at selected stages in an analysis by re-specifying SLIDELINE ASSIGNMENTS. However, assignments must remain unchanged for master and slave surface numbers, pre-contact detection, geometric definition and symmetrised Coulomb friction with the augmented method. When changing between frictional and other types of slideline the SLIDELINE PROPERTIES command will also be required to modify the friction coefficient.



7. The pre-contact detection flag, **Pcont**, is used to overcome problems encountered when applying an initial load to a discrete body which would be subjected to

unrestrained rigid body motion. This procedure is only applicable to static analyses and it is required when an initial gap exists between the slideline surfaces and a loading type other than PDSP or TPDSP is to be applied. When this facility is invoked the surfaces of the slidelines are brought together using interface forces (which act at right angles to the surfaces) and the applied loading. One of the surfaces must be defined on a discrete part of the structure which is free to move as a rigid body and the direction of movement is dictated by the interface forces, applied loading and support conditions. Incorrect use of this procedure could lead to initial straining in the bodies or to an undesired starting configuration. By selecting specific slidelines for the pre-contact process (i.e. slidelines where initial contact is expected) minimum initial straining will occur and more control over the direction of rigid body movement can be exercised. In the example above pre-contact is defined for slideline 1 but not for slideline 2. The force used to bring the bodies together (in addition to the applied loading) is computed from the product of the slideline stiffness and gap distance between the surfaces. This force can be factored using the SYSTEM parameter **SLSTPC**, (default=1.0). It is possible to define all slidelines for pre-contact and to specify a small value for **SLSTPC** (typically 1E-6), however, the initial applied loading must be small so that unrestrained rigid body motion does not occur and in general this approach tends to be less stable and is not recommended. It should be noted that if this procedure is used correctly any initial straining caused by the pre-contact process will disappear after the iterations for the first increment have lead to convergence.

8. Contact cushioning is designed to remove discontinuities in force and stiffness between in and out of contact states via an exponential function, thus aiming to reduce associated problems such as *chatter*. It can only be specified for general sliding with and without friction slideline types.

Compared to the basic penalty method, however, contact cushioning regards every node on every slideline as being active and may therefore increase processing time depending upon the number of slidelines and slideline nodes. If it is possible to make an assessment of potential areas where chatter may occur in a problem, contact cushioning should be used selectively on slidelines in those areas, although there is no harm in specifying it for all valid slidelines.

Contact cushioning uses an estimate of the normal contact force scalar in the exponential function. Where it is felt that the estimate of the normal contact force scalar is going to be consistently computed as either being too high or too low, the SYSTEM parameter **SLFNCS** (default = 1.0) can be used to factor those estimates.

9. Number of contact check passes is controlled by **islpss**. Currently this value only functions for rigid surfaces. For non-rigid surfaces, the default value of 2 will be used.

10. If **islpss** is set to one for a rigid surface, only the penetration of deformable surface into the rigid surface is prevented (and not visa versa).

11. Because rigid surfaces can not contact each other, only one of the slideline surfaces in a slideline assignment can be a rigid surface.

12. Curved slidelines (**islgem**=2) are defined using quadratic patches. These use information from a patch of slideline segments to define the curved geometry, while the contact forces are distributed to the closest segment. The quadratic patches and the curved geometry are set-up automatically within LUSAS Solver with no additional user specification. The standard patch configuration consists of two linear segments in 2D and four bi-linear segments (quadrilateral or triangular) in 3D. Where a patch definition is not possible the standard linear/bi-linear definition is used instead.
13. The quadratic patch formulation has a non-symmetric tangent stiffness matrix. The non-symmetric solver is therefore set automatically by LUSAS Solver.
14. Large mesh 'bias' with quadratic patches should be avoided to ensure a reasonable curve fit of the curved contact surface.
15. The standard slideline formulation applies an equal weighting to the force and stiffness at all contact nodes. With **isldst**=1, the weighting is based on the contacted area, accounting for effects at the edge of the contact area and for a non-uniform mesh.
16. The augmented Lagrangian method uses both penalty parameters and Lagrangian multipliers to reduce contact penetrations to below the tolerance specified under SLIDELINE PROPERTIES. The Lagrangian multipliers are introduced without increasing the number of variables in the solution. If at convergence the penetrations are not acceptable, the Lagrangian multipliers are updated using the penalty parameters, and the solution is rerun. This process is repeated until either the penetrations are acceptable or the maximum updates is reached (See *Nonlinear Control*).

    An augmented Lagrangian solution may therefore take longer than the standard penalty solution, but will be less sensitive to the penalty parameter and does not generate additional variables from the Lagrangian multipliers.
17. If the augmented Lagrangian solution is not reducing penetrations quickly enough, the penalty update procedure can be used to increase the penalty parameters (contact stiffnesses) in order to accelerate the reduction. Details of the procedure can be found under SLIDELINE PROPERTIES. The procedure only works with the augmented Lagrangian method.
18. The augmented Lagrangian criterion for determining whether a node is in contact is based on the sign of the Lagrangian multiplier rather than whether a node has penetrated. A positive value means a node is under tension and must therefore be out of contact, while a negative values means the node is under compression and therefore in contact. There are two techniques for employing this criterion. With Powell's method, the status of a node is only updated at the end of a Lagrangian loop, while with Uzawa's method the status is updated during each iteration.
19. The standard penalty and augmented Lagrangian formulations for Coulomb friction involve a non-symmetric stiffness matrix. Specifying **alsymf**=1 invokes a formulation that generates a symmetric stiffness matrix. This uses the Lagrangian multiplier from the previous update, rather than the current normal force, when

computing the frictional force. Although the solution is quicker, it can lag behind the non-symmetric version by one Lagrangian update.

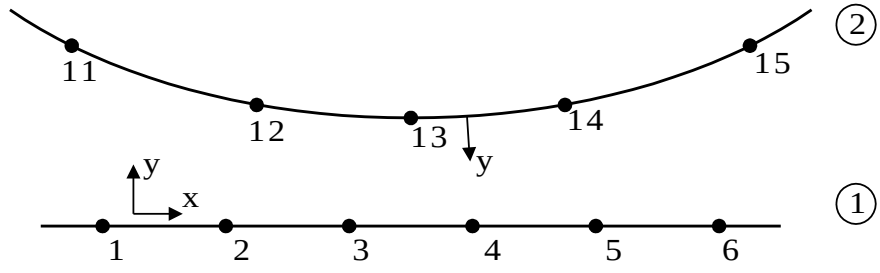
Example 1. 2D Slideline Surface Definition

Without Option 61:

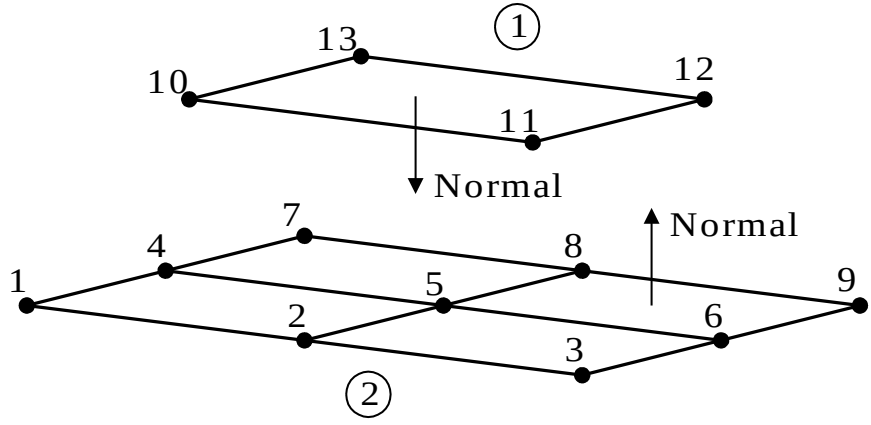
```
SLIDELINE PROPERTIES  1
  1 D D D D 0
SLIDELINE DEFINITION  1
 FIRST   1   2
   INC   1   1   5
SLIDELINE DEFINITION  2
 FIRST   11  12
   INC   1   1   4
SLIDELINE ASSIGNMENT
  1 1 2 1 D
```

With Option 61:

```
SLIDELINE PROPERTIES  1
  1 D D D D 0
SLIDELINE DEFINITION  1
 FIRST   1   2
   INC   1   1   5
SLIDELINE DEFINITION  2
 FIRST   12  11
   INC   1   1   4
SLIDELINE ASSIGNMENT
  1 1 2 1 D
```



Example 2. 3D Slideline Surface Definition




```
SLIDELINE PROPERTIES  1
  1 D D D D 0
SLIDELINE DEFINITION  2
FIRST   1   2   5   4
  INC   1   1   1   1   2
  INC   3   3   3   3   2
SLIDELINE DEFINITION  1
  10 13 12 11
SLIDELINE ASSIGNMENT
  1 2 1 1 D
```

Slideline Assignments Materials

The data section SLIDELINE ASSIGNMENTS MATERIALS complements the SLIDELINE ASSIGNMENT command but is used specifically with automatically generated surfaces for which a slideline surface number is not available.

SLIDELINE ASSIGNMENTS MATERIALS [TITLE title]

numsl IMAT1 IMAT2 isprop isltyp [P_{cont} islcsh islpss islgem]

numsl The slideline number

IMAT1 The master slideline surface material identifier IMAT

IMAT2 The slave slideline surface material identifier IMAT

isprop Slideline property number specified in SLIDELINE PROPERTIES data chapter

isltyp Type of slideline treatment required for this surface :-
- =0 for a null slideline
- =1 for general sliding without friction (default)
- =2 for general sliding with friction
- =3 for tied slidelines
- =4 for sliding only (without friction or lift off)

P_{cont} Pre-contact detection flag :-

no pre-contact process	= 0 (default)
pre-contact required	= 1

islcsh Contact cushioning parameter

No contact cushioning	= 0 (default)
Contact cushioning required	= 1

islpss Number of contact check passes (see Notes) :-

Two pass	= 2 (default)
One pass	= 1

Islgem Contact surface geometric definition : = 1

Notes

1. Automatically generated contact surfaces are not formally identified prior to solving the problem. In the absence of a specific surface identifier contact conditions are specified using the material identifiers IMAT corresponding to the elements on the contacted and contacting surface.
2. A node may come in contact with more than one surface but a contact element is only formed with the surface for which the contact node has penetrated least.
3. The contact surface geometry must be linear (**islgem** = 1), i.e. curved quadratic patches (**islgem** = 2) are not currently supported.
4. All parameters are as defined in SLIDELINE ASSIGNMENTS. Further details should are given in this data chapter.

Thermal Surfaces

Heat flow across the gap between two surfaces is modelled in LUSAS by the specification of links which define both the path of the heat flow as well as the gap's conductivity. The thermal surface data chapter describes a framework in which these links can be automatically generated and updated. In addition to direct heat flow between two points, defined by a link, indirect heat flows are possible by the formation of a link to an environmental node. Heat may also be lost directly to the general environment.

Heat Transfer by Conduction and Convection

The surfaces are discretised as a series of segments defined by the external nodes of the structure. In generating the links, heat flowing through an area of each segment is "lumped" to each of the segment nodes. The contributing portion of the segment surface assigned to a node is defined, for the following discussion, as a nodal area. It is then assumed that this heat flows in a direction normal to the nodal area. The origin of the normal flow vector is taken to be at a point one quarter the distance from the node to the segment centroid as this more fairly represents the source of the flow as being from an area rather than a node. To determine the heat flow from a nodal area to an opposing segment, the intercept of its nodal normal with that segment is sought. The size of the gap is defined as the magnitude of the intercept vector.

Environmental nodes may be used to represent the medium which separates the thermal surfaces between which heat is flowing. As the length of a link directly connecting two surfaces increases, the validity of the assumed flow becomes more tenuous. Alternatively, instead of forming a link, heat could flow directly to the surroundings, but in this case, the heat is lost from the solution. This, in some cases, is a poor approximation to reality, particularly when the thermal surfaces form an enclosure. In this instance an environmental node can be used to model the intervening medium, with all nodal areas which are not directly linked to other areas linked to the environmental node. The environmental node then re-distributes heat from the hotter surfaces of the enclosure to the cooler ones without defining the exact process of the transfer.

More than one environmental node may be defined and they may be connected together using links so that a better approximation to the behaviour of medium may be obtained.

The precise form of the link depends on the specified data, with the possibility of links forming between two surfaces, or a segment forming a link to either an environmental node or to the general environment, or finally, the surface may be considered to be insulated without any associated links. However, the order of priority of formulation of link type is well defined, and is set out below (the commands, which are written in capitals, are detailed in subsequent sections):

- For nodal area normals which have intercepted another surface segment:
  a. If a nodal area has penetrated another surface and THERMAL CONTACT PROPERTIES have been specified a link is formed using these properties.
  b. If a nodal area has penetrated another surface and no THERMAL CONTACT PROPERTIES have been specified and if THERMAL GAP PROPERTIES LINEAR or THERMAL GAP PROPERTIES GENERAL have been specified then these properties are used.
  c. If no penetration has occurred and if THERMAL GAP PROPERTIES LINEAR or THERMAL GAP PROPERTIES GENERAL have been specified then these properties are used.

d. If the length of the normal vector is less than or equal to the maximum permissible gap size defined by the properties in **b.** or **c.** then a link is formed otherwise it is considered that the surfaces are out of the range of influence and no link is formed.

- For nodal areas not linked to a surface segment by not having met conditions a. to d. above or not having intercepted another segment, and with THERMAL ENVIRONMENTAL PROPERTIES assigned:
  e. If the nodal area is associated to an environmental node then a link is formed to the node.
  f. If the nodal area is not associated to an environmental node, a link is formed to the general environment.
- If none of the conditions **a.** to **f.** apply then no link is formed.

Updated geometry is calculated in a structural analysis and is transferred to the thermal analysis using the thermo-mechanical coupling commands. If contact pressures are required, a slideline analysis must be run.

- For conduction and convection the thermal surface input requires the following data:
  a. Specification of gap properties using the commands
     THERMAL GAP PROPERTIES LINEAR
     THERMAL GAP PROPERTIES GENERAL
     THERMAL CONTACT PROPERTIES
  b. Specification of thermal surface segments using the command
     THERMAL_SURFACE DEFINITION
  c. Assignment of both surfaces and gap properties to a gap using the commands
     THERMAL ASSIGNMENT
     GAP SURFACE DEFINITION

  Additionally, the following data may also be prescribed:
  d. Environmental surface properties (these define the conductivity between a surface and the environment) using the commands
     THERMAL ENVIRONMENT PROPERTIES
     THERMAL_SURFACE PROPERTY ASSIGNMENT
  e. Environmental nodes may be defined and assigned to thermal surfaces using the commands:
     ENVIRONMENTAL NODE DEFINITION
     ENVIRONMENTAL NODE ASSIGNMENT

  and are effective if THERMAL ENVIRONMENT PROPERTIES and THERMAL_ SURFACE PROPERTY ASSIGNMENTS have been specified.

Heat Transfer by Diffuse Radiation

Heat transfer by diffuse radiation is modelled by defining the thermal surfaces that are engaged in radiative exchange. The thermal surfaces are then assigned to a radiation surface. The radiation surface defines those thermal surfaces that will exchange heat by radiation, since radiative heat transfer will only take place between thermal surfaces that define the same radiation surface. The geometry of the thermal surfaces is used to calculate the view factors between the segments that define the thermal surfaces.

A view factor between two segments expresses the fraction of radiative energy leaving one segment that is directly incident on the other segment. If two segments do not have a direct view of each other, for example they may face in opposite directions or there may be other segments blocking the direct view, then the view factor is zero. However two segments with a zero view factor may still exchange heat by reflection off other segments in the thermal surfaces defining the complete radiation surface. If the thermal surfaces defining a radiation surface form an enclosure then the sum of the view factors for any surface segment must be 1.0. The view factors for each radiation surface are calculated in a data check run as well as a full solution run. All the thermal surfaces are assumed to be opaque so that the reflectivity is given by (1.0- emissivity) and the emissivity may be input by the user and assigned to each thermal surface.

The emissivity is a fraction between 0.0 and 1.0 and describes the emissive power of a surface as a fraction of the blackbody emissive power. The blackbody emissive power is given by $\sigma T4$, where

σ = Stefan Boltzmann constant = 5.6697e-8 W/m2/K4

T = temperature of the body in Kelvin

If you are working in units other than Watts and metres then the Stefan-Boltzmann constant will need to be set using the SYSTEM parameter **STEFAN**. By default all temperature data input and output by LUSAS will be in Kelvin, however you can choose to work in Celsius (Centigrade) by using Option 242. The temperature units of the Stefan Boltzmann constant must always be input in Kelvin.

Using the view factors and emissivity data LUSAS can assemble a matrix that describes all the possible paths for heat to be radiated around the radiation surface. If the radiation surface does not form an enclosure then the segments may optionally radiate to the environment as well as each to other.

If you have a symmetric problem in which the plane(s) of symmetry cut through a radiation surface then it is necessary to define the plane(s) of symmetry. This enables LUSAS to impose the zero flux condition necessary along a plane of symmetry in a thermal model.

For diffuse radiation the thermal surface input requires the following data:

1. Specification of surface radiation properties using the command:

THERMAL RADIATION PROPERTIES

2. Specification of thermal surface segments using the command:
   THERMAL_SURFACE DEFINITION
3. Assignment of radiation properties to thermal surfaces using the command:
   THERMAL_SURFACE PROPERTY ASSIGNMENT
4. Assignment of thermal surfaces to form a radiation surface using the commands:
   THERMAL ASSIGNMENT
   RADIATION SURFACE

Additionally the following data may be specified if the model contains planes of symmetry that cut through thermal surfaces defining a radiation surface:

1. Specification of planes of symmetry using the command:
   THERMAL RADIATION SYMMETRY
2. Assignment of symmetry planes to a symmetry surface using the command:
   THERMAL_SURFACE SYMMETRY ASSIGNMENT
3. Assignment of a symmetry surface to a radiation surface using the commands:
   THERMAL ASSIGNMENT
   RADIATION SURFACE

Thermal Gap Properties Linear

The data section THERMAL GAP PROPERTIES LINEAR is used to define thermal properties which may take a linear variation across a gap opening. It is possible to redefine the properties as an analysis progresses, however, the properties must also be reassigned using the THERMAL ASSIGNMENT data chapter.

THERMAL GAP PROPERTIES LINEAR [TITLE title]

gmat k_o [h_{co} dk/dl dh/dl L_{max} T]

gmat	Gap properties identifier
ko	Gap conductance for a closed gap
hco	Convective heat transfer coefficient for a closed gap
dk/dL	Variation of gap conductance with gap opening
dh/dL	Variation of convective heat transfer coefficient with gap opening
Lmax	Maximum link length beyond which a link will not be created
T	Reference temperature

Notes

1. If the gap opening is such that a negative gap conductance or convective coefficient is calculated, then that gap property is set to zero.
2. The maximum length, **Lmax**, limits the range of validity of the link element to which these gap properties will be applied. If the length of the link is calculated to be greater than **Lmax**, then heat will flow either to an environmental node, to the

general environment or the surface will be considered to be insulated, depending on the ENVIRONMENTAL SURFACE PROPERTIES input.

3. Gap openings are calculated by evaluating the intersection of the surface normal, originating from a point 1/4 the distance from a segment node to the segment centroid, with an opposing surface.
4. Gap properties are assigned to a gap using the THERMAL ASSIGNMENT data chapter.

Thermal Gap Properties General

The data section THERMAL GAP PROPERTIES GENERAL is used to define thermal properties which can vary according to the size of the gap opening. It is possible to redefine the properties as an analysis progresses, however, the properties must also be reassigned using the THERMAL ASSIGNMENT data chapter.

THERMAL GAP PROPERTIES GENERAL [TITLE title]

gmat $<$ $\mathbf{k_i}$ $\mathbf{h_{ci}}$ $\mathbf{L_i}$ $>_{\mathbf{i=1,n}}$ **T**

n	Number of gap openings for which properties will be defined
gmat	Gap properties identifier
ki	Gap conductance for gap opening **Li**
hci	Convective heat transfer coefficient for gap opening **Li**
Li	Size of the ith gap opening
T	Reference temperature

Notes

1. **Lmax** and **Lmin** are defined as the maximum and minimum of **Li** (**i**=1,**n**).
2. If the actual gap opening is less than the minimum value of gap opening, **Lmin**, the gap properties are assumed to be those corresponding to **Lmin**. If the gap opening is such that a negative gap conductance or convective coefficient is calculated, then that gap property is set to zero.
3. If the actual gap opening is greater than the maximum value of gap opening, **Lmax**, then heat will flow either to an environmental node, or to the general environment or the surface will be considered to be insulated, depending on the ENVIRONMENTAL SURFACE PROPERTIES input.
4. If only 1 point is input, **n**=1, **Lmax** is taken to be **L1** and the gap properties are constant over a gap opening of 0 to **L1**.
5. Gap openings are calculated by evaluating the intersection of the surface normal, originating from a point 1/4 the distance from a segment node to the segment centroid, with an opposing surface.
6. Gap properties are assigned to a gap using the THERMAL ASSIGNMENT data chapter.

Thermal Contact Properties

The data section THERMAL CONTACT PROPERTIES is used to define thermal properties which can vary according to the surface pressure between two contacting surfaces. The surface pressures must be evaluated using the slideline facility and introduced to the thermal analysis via thermo-mechanical coupling. It is possible to redefine the properties as an analysis progresses, however, the properties must also be reassigned using the THERMAL ASSIGNMENT data chapter.

THERMAL CONTACT PROPERTIES n [TITLE title]

cmat < k_i h_{ci} P_i >$_{i=1,n}$ T

n	Number of contact pressures for which properties will be defined
cmat	Contact properties identifier
ki	Gap conductance for contact pressure **Pi**
hci	Convective heat transfer coefficient for contact pressure **Pi**
Pi	The ith contact pressure defined
T	Reference temperature

Notes

1. **Pmax** and **Pmin** are defined as the maximum and minimum of **Pi** (**i**=1,**n**).
2. If the surface pressure lies outside the range bounded by **Pmax** and **Pmin**, then the value taken is **Pmin** if the surface pressure is less than **Pmin** and correspondingly the value of **Pmax** is taken if the surface pressure exceeds **Pmax**. If the gap opening is such that a negative gap conductance or convective coefficient is calculated, then that gap property is set to zero.
3. Contact pressures can only be calculated using contact forces evaluated from a slideline analysis. The thermo-mechanical coupling option must be used to transfer this data to the thermal analysis.
4. Contact forces in a slideline analysis are computed if a node penetrates the opposing surface, whilst in a thermal analysis a contact area is calculated if a point 1/4 the distance from the node to the segment centroid has penetrated the opposing surface. This can occasionally lead to different contact predictions between the analyses, particularly at the ends of surfaces or if a sharp surface impacts on a smooth surface.
5. Contact pressure is calculated by dividing the nodal force by the area of all adjacent surfaces which are in contact with opposing surfaces.
6. If a surface is not in contact, a thermal link will be established either to an opposing surface, an environmental node, the general environment or the surface may be considered to be insulated, depending on the ENVIRONMENTAL SURFACE PROPERTIES input.
7. Contact properties are assigned to a gap using the THERMAL ASSIGNMENT data chapter.

Thermal Radiation Properties

The data section THERMAL RADIATION PROPERTIES is used to define the emissivity of thermal surfaces engaged in heat transfer by radiation.

```
THERMAL RADIATION PROPERTIES  [TITLE  title]

rmat  ε  T
```

rmat	Radiation property identifier
ε	Emissivity
T	Reference temperature

Notes

1. All surfaces are diffuse and opaque so the reflectivity is given by (1-emissivity).
2. The radiation properties are assigned to a thermal surface using the THERMAL_SURFACE PROPERTY ASSIGNMENT data chapter.
3. By default all temperatures input and output in heat transfer problems involving radiation will be in Kelvin. Option 242 may be set to allow temperatures to be input and output in Celsius (Centigrade).
4. Option 131 cannot be used with radiation properties. This is because Option 131 relies on a symmetric matrix and radiation is non-symmetric.

Thermal Environment Properties

The data section THERMAL ENVIRONMENT PROPERTIES is used to define the heat transfer coefficients for the interface of a surface and a bounding medium. The thermal environmental properties are properties of a surface, not a gap, and are assigned directly to a surface using the THERMAL_SURFACE PROPERTY ASSIGNMENT data chapter. Properties may be redefined as an analysis progresses, however, to invoke the redefinition the properties must also be reassigned.

```
THERMAL ENVIRONMENT PROPERTIES  [TITLE  title]

emat  k  [hc [εenv] envtmp T]
```

emat	Environment properties identifier
k	Surface conductance to environment
hc	Surface convective heat transfer coefficient
εenv	Environment emissivity (see *Notes*)
envtmp	Environmental temperature
T	Reference temperature

Notes

1. A surface segment will try to establish a link with another surface segment. If it fails to do so, heat can flow to an environmental node or to the general environment. The conduction properties for this flow are defined using

THERMAL ENVIRONMENT PROPERTIES. If these properties are not assigned to a surface and a link is not formed, and the surface does not define a RADIATION SURFACE, then surface is treated as though it is insulated.

2. If the surface has an associated environmental node, a link is established to this node with a conductivity defined using the THERMAL ENVIRONMENT PROPERTIES; the environmental temperature, **envtmp**, is ignored. If there is not an associated environmental node, heat is lost directly to the environment using a conductivity defined using the THERMAL ENVIRONMENT PROPERTIES; the environment takes a temperature of **envtmp**.
3. To specify the environment emissivity Option 253 must be set. The environment emissivity is only used if a segment within a RADIATION SURFACE has a view factor to the environment that exceeds a defined tolerance. The view factor to the environment for a surface segment is defined by 1.0 - (sum of the view factors from the segment to all the other segments defining the RADIATION SURFACE). The tolerance may be set by the system variable **TOLFIJ**.
4. Thermal environment properties are assigned to a surface using the THERMAL_SURFACE PROPERTY ASSIGNMENT data chapter.

Thermal Radiation Symmetry

The data section THERMAL RADIATION SYMMETRY is used to define lines or planes of symmetry that cut through radiation surfaces.

```
THERMAL RADIATION SYMMETRY  [TITLE  title]
LINE  nsym  {NODES  N1 N2 | EQUATION  LX LY L}
PLANE  nsym  {NODES  N1 N1 N3 | EQUATION  CX CY CZ C}
```

nsym Symmetry line or plane identifier
N1, N2, N3 Node numbers defining a line or plane of symmetry
LX, LY, L Coefficients in the line equation **LX**.x + **LY**.y = **L**
CX, CY, CZ, C Coefficients in the plane equation **CX**.x + **CY**.y + **CZ**.z = **C**

Notes

1. A line of symmetry must always lie in the global XY plane of the model.
2. A line of symmetry may be used in either a 2D, axisymmetric or 3D analysis. In an axisymmetric analysis a line of symmetry must be perpendicular to the axis about which the model is spun. In a 3D analysis a line of symmetry is converted to a plane of symmetry passing through the defined line and parallel to the global Z axis of the model.
3. A plane of symmetry may be used in either a 2D, axisymmetric or 3D analysis. In a 2D analysis the plane of symmetry must be parallel to the global Z axis of the model. In an axisymmetric analysis a plane of symmetry must be perpendicular to the axis about which the model is spun and parallel to the global Z axis of the model.

4. Radiation symmetry lines and planes only need to be defined when the plane(s) of symmetry in a model cuts through a radiation surface. Symmetry planes are not required if no radiation surfaces are defined.
5. The symmetry planes are assigned to a symmetry surface using the THERMAL_SURFACE SYMMETRY ASSIGNMENT data chapter.

Thermal Surface Symmetry Assignment

The data section THERMAL_SURFACE SYMMETRY ASSIGNMENT is used to assign THERMAL RADIATION SYMMETRY lines and planes to a symmetry surface.

THERMAL_SURFACE SYMMETRY ASSIGNMENT isym [TITLE title]

< nsym$_i$ >$_{i=1,npln}$

isym Symmetry surface identifier
npln Number of lines and/or planes of symmetry defining symmetry surface
nsym **npln** symmetry line and/or plane identifiers defining symmetry surface

Notes

1. The number of symmetry planes assigned to a symmetry surface must be sufficient to enable LUSAS to generate the full radiation surface by mirroring the defining thermal surfaces successively in each symmetry plane in the order in which the symmetry identifiers are specified in the assignment command.
2. The symmetry surface is assigned to a radiation surface using the RADIATION SURFACE data chapter.

Thermal Surface Definition

The data section THERMAL_SURFACE DEFINITION is used to define the topology of each thermal surface. It is possible to redefine the surfaces as an analysis progresses, however, the surfaces must also be reassigned using the THERMAL GAP ASSIGNMENT or the THERMAL_SURFACE PROPERTY ASSIGNMENT data chapters.

THERMAL_SURFACE DEFINITION ntsurf [TITLE title]

< Ni >i=1,nseg

ntsurf Thermal surface identifier
nseg Number of nodes defining each segment
Ni **nseg** node numbers defining segment

Notes

1. A segment is defined as an element face which forms part of the thermal surface. Two nodes are required to define the surface for 2D analyses. Three or four nodes are required to define a segment for a 3D problem, depending on the particular element face. The segments must be located along the object boundary with segment node ordering defined in a consistent direction, however, segments may be specified in an arbitrary order.
2. The heat flow from a surface is along a vector which is normal to the surface of the segment. For segments in two dimensions, the normal is defined using a right-hand screw rule with the local x-axis running from the first node to the second, the y-axis defines the surface normal with the z-axis coming out of the problem plane. For 3D surfaces, the node definition for surface segments must be labelled in an anti-clockwise direction (when looking towards the structure along the outward normal to the thermal surface).
3. If two 2D elements of different thickness are connected to a segment node, the thickness of the thermal segment is taken as the average of the two element thicknesses.
4. Thermal surface definitions are assigned using the THERMAL ASSIGNMENT data chapter.

Thermal Surface Property Assignments

The data section THERMAL_SURFACE PROPERTY ASSIGNMENTS assigns the THERMAL ENVIRONMENT PROPERTIES and THERMAL RADIATION PROPERTITES to a thermal surface previously defined using the THERMAL_SURFACE DEFINITION command.

THERMAL_SURFACE PROPERTY ASSIGNMENT [TITLE title]

ntsurf emat rmat [npnts]

ntsurf	Thermal surface identifier
emat	Thermal environment properties identifier
rmat	Thermal radiation properties identifier
npnts	Number of sub-segments to use in the calculation of radiation view factors. A default value of 2 will be used if not input by the user and if the thermal surface defines a radiation surface, otherwise default=0

Notes

1. THERMAL ENVIRONMENT PROPERTIES are not mandatory. If specified then they are used by surface segments that do not form a conduction or convection link to another surface segment, or they may be used by a thermal surface that defines a radiation surface to establish a radiation link to the environment depending on the surface segment environment view factor.
2. THERMAL RADIATION PROPERTIES are mandatory for any thermal surface that defines a radiation surface.

3. If Option 131 is specified when radiation surfaces are defined, the option is switched **off** and a warning message is issued. (Option 131 is switched on by default).
4. The number of sub-segments will default to a value of 2 if not specified for any thermal surfaces that define radiation surfaces. The number of sub-segments is used in the view factor calculations to sub-divide each surface segment into a series of smaller segments. The accuracy of the view factors will increase as more sub-segments are used but the calculation time will also increase. Try to run with the minimum number of sub-segments that give the required accuracy. The maximum number of sub-segments allowed in 2D and axisymmetric models is 50 and in 3D models the maximum allowed is 30.
5. THERMAL ENVIRONMENT PROPERTIES and the THERMAL_SURFACE DEFINITION may be redefined and reassigned at any point in the solution.

Thermal Assignment

The data section THERMAL ASSIGNMENT is used to define the thermal gaps, their surfaces, properties and processing, and thermal radiation surfaces.

THERMAL ASSIGNMENT [TITLE title]

GAP DEFINITION [UPDATE | NO_UPDATE] [NO_SHADING | SHADING] [ACTIVE | INACTIVE]

GAP PROPERTIES igap gmat [cmat]

GAP SURFACE DEFINITION igap ntsurf$_1$ [ntsurf$_2$]

RADIATION SURFACES irad isym [TITLE title]

< itsurf$_i$ >$_{i=1,nsurf}$

igap Thermal gap identifier
UPDATE Update geometry from coupled structural analysis (default)
NO_UPDATE Prevent processing of updated geometry from coupled structural analysis (default **UPDATE** geometry)
NO_SHADING No checks for shading of elements defining a thermal gap (default **SHADING**)
SHADING Check for shading of surface segments defining a thermal gap (default)
ACTIVE Process gap (default)
INACTIVE Suppress further processing of thermal surface (default **ACTIVE**)
gmat Gap property identifier (default=0, see *Notes*)
cmat Contact property identifier (default=0, see *Notes*)
ntsurf1 Thermal surface identifier (see *Notes*)
ntsurf2 Thermal surface identifier (see *Notes*)
irad Radiation surface identifier
isym Symmetry surface identifier
nsurf Number of thermal surfaces defining radiation surface

itsurf **nsurf** Thermal surfaces defining the radiation surface (see *Notes*)

Notes

1. Gaps may be defined and redefined at any point in the solution.
2. GAP PROPERTIES and GAP SURFACE DEFINITIONS may be re-assigned at any point in the solution.
3. The **NO_UPDATE** flag can be set to prevent the update and subsequent re-calculation of thermal links when geometry is read from a coupled structural analysis. The default is to **UPDATE** geometry.
4. The **SHADING** parameter forces a complete search of all possible segments. The nearest segment with which it is possible to form a link is then taken as the linkage segment. By default the parameter is enabled.
5. The **INACTIVE** command suspends further processing of the thermal gap, which has the effect of insulating flow across it. A gap may be re-activated at a later stage in the solution by re-issuing the GAP DEFINITION command with the **ACTIVE** parameter.
6. The use of the GAP DEFINITION command resets the **UPDATE**, **SHADING** and **ACTIVE** parameters to their default or defined values.
7. Either a gap property identifier, **gmat**, or a contact property identifier, **cmat**, must be specified if a gap, **igap**, is defined. Specification of a 0 or **D** for an identifier will indicate that no property identifier is to be used.
8. **ntsurf1**, **ntsurf2** and **itsurf** are the identifiers defined with the THERMAL_SURFACES DEFINITION data chapter.
9. At least one surface, **ntsurf1**, must be specified if a gap, **igap**, is defined. Links will then be set up between segments of this surface alone.
10. Only one thermal surface need be defined in a thermal gap. The specification of a second surface, **ntsurf2**, is for numerical efficiency. The search for linkage between the two defined segments commences with segments on the second surface; if no possible link is found, the first surface is then checked.
11. 0 or **D** should be input for the symmetry surface, **isym**, if the radiation surface, **irad**, has no symmetry assignment.
12. Any number of thermal surfaces may be used to define a radiation surface and the thermal surfaces do not have to form a continuous line/surface or enclosure.
13. Radiative heat transfer will only take place between thermal surfaces used in the definition of the same radiation surface.
14. The radiation view factors are only dependent on the geometry of the segments defining the radiation surface. So the view factors are only calculated on the very first iteration of a thermal analysis and are recalculated each time the nodal coordinates are updated in a thermal coupled analysis. Option 256 may be used to suppress the recalculation of the view factors in a coupled analysis.

15. By default all temperatures input and output in heat transfer problems involving radiation will be in Kelvin. Option 242 may be set to allow temperatures to be input and output in Celsius(Centigrade).

16. If radiation surfaces have been defined and the problem units are not Watts and metres then the Stefan Boltzmann constant must be set in the appropriate problem units using the system variable **STEFAN**.

Radiation Shading

The RADIATION SHADING data chapter allows the user to specify which of the thermal surfaces defining a radiation surface are shading surfaces, partially or fully blocking the view between other thermal surfaces in the radiation surface

```
RADIATION SHADING  irad  [TITLE  title]
< itshad >i=1,nshad
```

irad Radiation surface identifier

nshad Number of shading thermal surfaces

itshad Thermal surfaces that obstruct the view between the thermal surfaces that define the radiation surface.
itshad = -1: all thermal surfaces defining the radiation surface are assumed to obstruct the view between segments in the radiation surface.
itshad = 0: no thermal surfaces obstruct any view between any of the thermal surfaces defining the radiation surface, for example the inside of a cube.

Note. If the radiation shading chapter is omitted then all thermal surfaces defining the radiation surface are assumed to obstruct the view between segments in the radiation surface.

If you are in any doubt about shading within a radiation surface then the fully obstructed option, **itshad** = -1, should be used. This is because the correct view factors are obtained even if a thermal surface is declared as a shading surface when it is not, but a shading surface not declared will lead to incorrect view factors. However you should always try to correctly identify the shading surfaces as this does speed up the view factor calculation.

Radiation View Factor Read

The RADIATION VIEW FACTOR READ chapter allows you to specify which radiation surface view factors are to be read from a file.

```
RADIATION VIEW FACTOR READ  [TITLE  title]
irad
```

irad Radiation surface identifier

The filenames for the radiation surfaces must follow the convention:

jnnmm.rvf

where:

j = job name
nn = radiation surface counter, =01 for the first radiation surface defined, 02 for the second etc.
mm = file counter only required if the radiation surface view factors do not all fit in the default file size of 1Mbyte, =blank for the first file, 01 for the second file, etc.

For example, imagine job **radt** contains two radiation surfaces with identifiers 3 and 4, then the view factors would be read from files:

- **radt01.rvf**, **radt0101.rvf**, **radt0102.rvf** ... for radiation surface 3, and
- **radt02.rvf**, **radt0201.rvf**, **radt0202.rvf** ... for radiation surface 4

Radiation view factor file format

	Number of integer words		
Description	**Single Precision**	**Double Precision**	**Convex**
Geometry type 1 = axisymmetric 2 = 2D planar 3 = 3D	1	2	1
Number of segments	1	2	1
Calculation type = 1	1	2	1
(100*number of records/file) + number of bytes/integer	1	2	1
Number of bytes/real	1	2	1
Segment areas	N	2N	N
Direct interchange area matrix (Ai * Fij)	N*N	4*N*N	N*N

Note. Option 255 creates the view factor transfer files for each radiation surface if they do not already exist. These files are very useful if the job is re-run, because the addition of the RADIATION VIEW FACTOR READ chapter makes LUSAS read in the view factors from these files instead of repeating the view factor calculations.

View Factor Output

The data section VIEW_FACTOR OUTPUT is used to control the detail of view factor output in the LUSAS output file.

```
VIEW_FACTOR OUTPUT  [TITLE  title]
irad  irad_last  irad_diff  [iout]
```

irad iradlast

iraddiff The first, last and difference defining a series of radiation surfaces with identical output control

iout Radiation surface view factor output number

=0 no view factor output

=1 print the sum of the view factors for each surface segment in the radiation surface and, print the largest deviation in any segment sum from 1.0

=2 print all the individual view factors from each surface segment to all the other segments defining the radiation surface and, print the sum of the view factors for each surface segment in the radiation surface and, print the largest deviation in any segment sum from 1.0

=3 print just the largest deviation in any surface segment view factor sum from 1.0

Notes

1. The default control value for all radiation surfaces is 1.

Environmental Node Definition

The data section ENVIRONMENTAL NODE DEFINITION defines environmental nodes to be used with the thermal surfaces. They are used to distribute heat transferred to the medium separating the thermal surfaces. Heat will flow, via the medium, from hotter surfaces to cooler surfaces without the direct formation of thermal links between these surfaces. The medium is assumed to be of constant temperature as defined by the temperature of the environmental node.

```
ENVIRONMENTAL NODE DEFINITION  [TITLE  title]
encode  [k C a_r envtmp]
```

enode Environmental node number (see *Notes*).

k Conductance from the environmental node to an external environment of constant temperature.

C Specific heat capacity multiplied by the mass of the medium associated to the environmental node (used in transient thermal analysis).

ar Association radius. Segment nodes lying outside this radius are not associated to the environmental node (default = infinity, see *Notes*).

envtmp External environmental temperature.

Notes

1. Environmental nodes are assigned to gaps using ENVIRONMENTAL NODE ASSIGNMENTS command.
2. All environmental nodes must be defined in the initial data section prior to the first solution. Their node number and position are defined in NODE COORDINATES. Unassigned and disassociated environmental nodes will take the temperature of the external environment **envtmp**.
3. Environmental nodes may be connected together using other elements such as thermal links and bars. Care must be taken in ensuring that environmental nodes connected to links have the proper boundary conditions.
4. Environmental nodes may be assigned to more than one gap. More than one environmental node may be assigned to one gap.
5. Environmental nodes may be supported and temperatures prescribed using either the SUPPORT or PDSP commands.
6. Concentrated loads (i.e. thermal fluxes) may not be applied directly to environmental nodes. However, thermal bars may be connected to the environmental node and concentrated loads applied to the end of the thermal bar.
7. The associated radius will be ignored if this value is set to **D**. This effectively means that all nodes may be associated with this environmental node.
8. An example on the definition and usage of environmental nodes is given below.

Environmental Node Assignments

The data section ENVIRONMENTAL NODE ASSIGNMENTS assigns environmental nodes to thermal gaps.

ENVIRONMENTAL NODE ASSIGNMENTS [TITLE title]

N N_{last} N_{diff} igap

N Nlast Ndiff The first node, last node and difference between nodes of the series of environmental nodes.

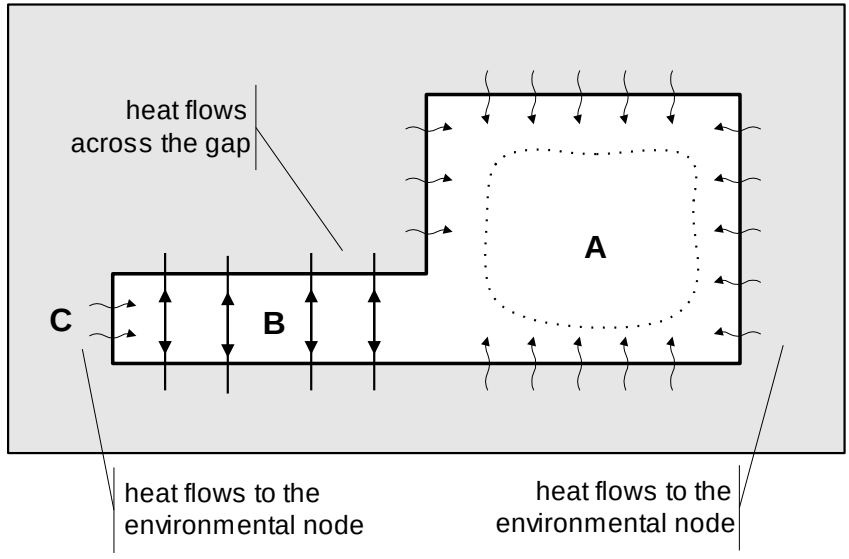
igap Identifier defining the gap to which the environmental nodes are to be assigned (see *Notes*).

Notes

1. All previous assignments are overwritten if the command ENVIRONMENTAL NODE ASSIGNMENTS is re-issued in the solution.
2. **igap** is defined in the THERMAL GAP DEFINITION data chapter.
3. An example of the assignment and usage of environmental nodes is given below.

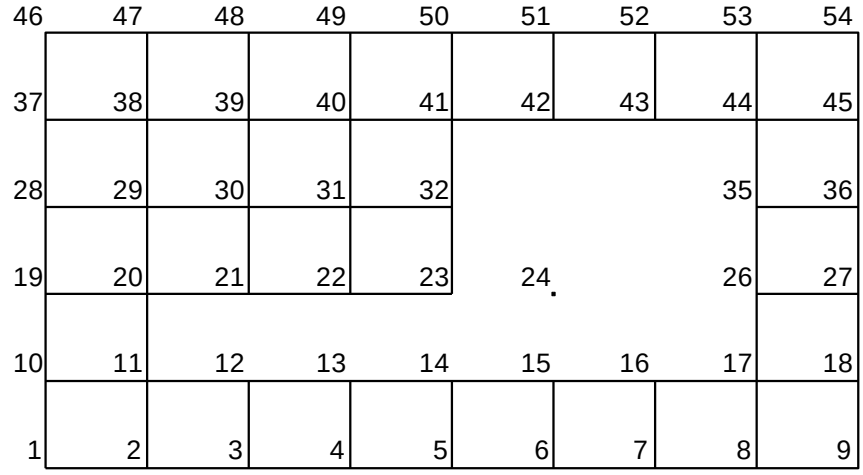
Example. Use of Environmental Nodes

Shown below is an enclosure with a possible distribution of heat flow:



Heat flows directly across the enclosure at its narrowest point B and via an environmental node at its widest point A. At C heat flow is to the environmental node, as the length of its link is beyond the maximum permitted.

Meshing the enclosure surround as shown below:



where node 24 is the environmental node. The thermal surface data would take the form:

```
     :
THERMAL GAP PROPERTIES LINEAR: define gap properties
 44 100 0 0 0 2 0
THERMAL ENVIRONMENT PROPERTIES: define surface/gas
interface properties
 66 50
THERMAL_SURFACE DEFINITION 88: define surface as inside
of enclosure
 FIRST 11 12
   INC  1   1  6
 FIRST 17 26
   INC  9   9  3
 FIRST 44 43
   INC -1 -1   3
 FIRST 41 32
   INC -9 -9   2
 FIRST 23 22
   INC -1 -1   3
 20 11
```

```
THERMAL_SURFACE PROPERTY ASSIGNMENTS: assign surface
properties to surface
 88 66
THERMAL GAP ASSIGNMENT: assign properties and surfaces
to gap 1
 GAP DEFINITION 1
 GAP PROPERTY   1  44
 GAP SURFACES DEFINITION    1  88
ENVIRONMENTAL NODE DEFINITION: define node 24 as
environmental node
 24
ENVIRONMENTAL NODE ASSIGNMENTS: assign environmental
node 24 to gap 1
 24 0  0    1
    :
```

Nodal Freedoms

The NODAL FREEDOMS data chapter optionally defines the freedoms at a node when using thick shell elements (TTS3, QTS4, TTS6, QTS8). This command allows five or six degrees of freedom to be specified for a node where either two 'local' or three global rotations will apply. This facility may be used in conjunction with TRANSFORMED FREEDOMS to specify loading or boundary conditions in more convenient directions.

NODAL FREEDOMS

N N_{last} N_{diff} n_{free}

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical nodal freedoms.

n_{free} The number of freedoms (must be 5 or 6)

Notes

1. Five degrees of freedom will automatically be assigned to a node unless:
   - Another type of element with six global degrees of freedom is connected to the same node.
   - The maximum angle between adjacent shell element normals at the node is greater than the SYSTEM parameter **SHLANG** (default = 20°).
2. If six degrees of freedom are specified for a node, care should be taken that the rotation about the element normal is restrained IF NECESSARY to prevent

singularities. Circumstances in which singularities may occur if this rotation is not restrained are:

- When only one element is connected to the node.
- When the surface modelled by the elements is quite flat.

3. It is recommended that five degrees of freedom are used whenever possible.
4. A description of the two 'local' rotations is given in the *LUSAS Element Reference Manual*.

Freedom Template

The FREEDOM TEMPLATE command optionally defines the list of freedoms for which values are defined in SUPPORT CONDITIONS, CONSTRAINT EQUATIONS, RETAINED FREEDOMS, MODAL SUPPORTS and LOADCASE data chapters.

FREEDOM TEMPLATE

< fretyp$_{(i)}$ **>**$_{i=1,nfrtmp}$

fretyp$_{(i)}$ Freedom type for each freedom in template.
nfrtmp Number of freedoms in template.
Valid freedom types are:

U - displacement in global X-direction.
V - displacement in global Y-direction.
W - displacement in global Z-direction.
THX - rotation about the global X-axis.
THY - rotation about the global Y-axis.
THZ - rotation about the global Z-axis.
THL1 - local rotation about the first loof point.
THL2 - local rotation about the second loof point.
DU - hierarchical displacement
DTHX - hierarchical local rotation
PHI - field variable
THA - rotation about first local axis for thick shells
THB - rotation about second local axis for thick shells

Notes

1. If FREEDOM TEMPLATE is not specified then the values input on the SUPPORT CONDITIONS, CONSTRAINT EQUATIONS, RETAINED FREEDOMS and LOADCASE will be applied to the freedoms at the node in the order that they occur.
2. The FREEDOM TEMPLATE command must be input when MODAL SUPPORTS are specified.
3. The following LOADCASE options utilise the FREEDOM TEMPLATE information:
   - TPDSP, PDSP, CL, VELOCITY, ACCELERATION

4. If superelements are used without any standard elements the FREEDOM TEMPLATE command must be specified.
5. If superelements are used that have more freedoms than the standard elements, and values are to be prescribed for these freedoms, then the FREEDOM TEMPLATE must be specified.

Example. Freedom Templates

If an analysis uses 3D beam elements with freedoms U, V, W, THX, THY, THZ and the template freedoms are defined as:

```
FREEDOM TEMPLATE
U   V   THZ
```

Then the support data would be of the form:

```
SUPPORT CONDITIONS
1  1  0     R  F  R
```

which defines U and THZ as fixed and V as free; the other freedoms (i.e. W, THX, THY) will be considered as free.

Cartesian Sets

Cartesian coordinate sets may be used to define a set of local xyz-axes relative to the global axes. The data section CARTESIAN SETS is used to define the required local Cartesian coordinate axes.

```
CARTESIAN SETS  [CYLINDRICAL]  [TITLE  title]
nset  N0 [N1 N2]

CARTESIAN SETS  MATRIX  [TITLE  title]
nset  < Ri >i=1,nmatrix
```

nset The Cartesian set identification number.
N_0 The node defining the Cartesian set origin.
N_1 Additional node required to uniquely define set.
N_2 Additional node required to uniquely define set.
R_i The matrix terms (row by row) defining the Cartesian set transformation (see *Notes*).
nmatrix The number of terms in the Cartesian set transformation matrix (4 for 2D problems and 9 for 3D problems).

Notes

1. The nodes required to define a Cartesian set are as follows (see figures below):

- ❑ 2D problem:
  - The N1 node defines the local x-axis.
  - The local y-axis is determined using the right-hand screw rule, with the z-axis coming out of the plane of the mesh
- ❑ 3D problem:
  - The N1 node defines the local x-axis.
  - The N2 node is any point lying in the positive quadrant of the local xy-plane.
  - The local z-axis is defined using the right-hand screw rule.

2. Cylindrical Cartesian sets define the radial, tangent and normal axes transformations. The tangent vector is positive in the direction of a clockwise rotation when looking along the local normal axis from the origin N0. The local axes are evaluated at an arbitrary point within the domain. The nodes used to define a Cylindrical Cartesian set are as follows (see figures below):

- ❑ 2D problem:
  - No additional data is required.
  - The local normal axis is directed out of the plane of the mesh.
  - If a node is coincident with N0 the radial axis will coincide with the x-axis and the tangential axis will coincide with the y-axis.
- ❑ 3D problem:
  - The N1 node defines the axis from which the cylindrical vectors are evaluated.
  - The local normal axis is defined as being positive in the direction from N0 to N1.
  - If a node lies on the local normal axis, by 1 default the radial axis will lie in the xy-plane and the tangential axis will complete the corresponding right-hand coordinate system.
  - Order of axes r´, t´, n´.

3. The Cylindrical Cartesian set is evaluated at the centre point of an element when determining material properties. The centre point is defined as the average of the nodal x, y and z coordinates.

4. The Cartesian set transformation matrix defines the 2D or 3D transformation from local to global coordinate systems as:

$$\{U\} = [R]^T \{u'\}$$

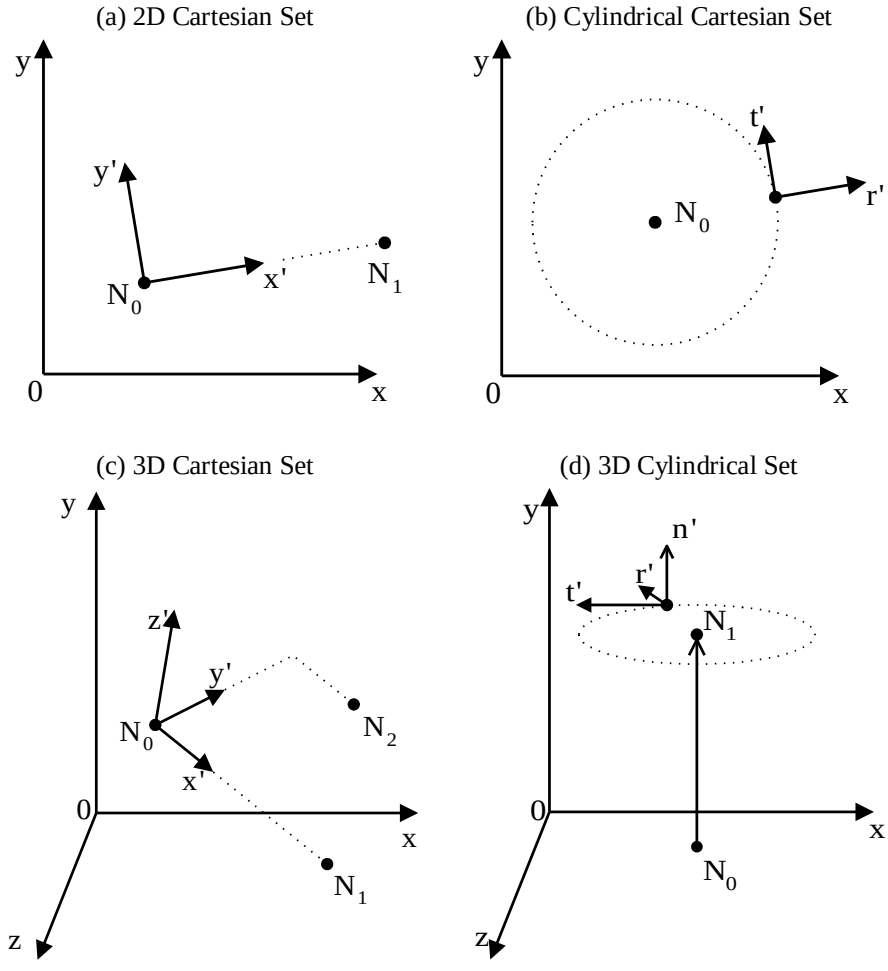
where {u´} is a vector in the local coordinate system and {U} is the vector in the global coordinate system. [R] is the transformation matrix. The dimensions of [R] are 2x2 for 2D problems and 3x3 for 3D problems. [R] may be constructed by defining the orthogonal vector directions which define the local coordinate directions. For example:

$$[R]^T = [\{x'\}, \{y'\}, \{z'\}]$$

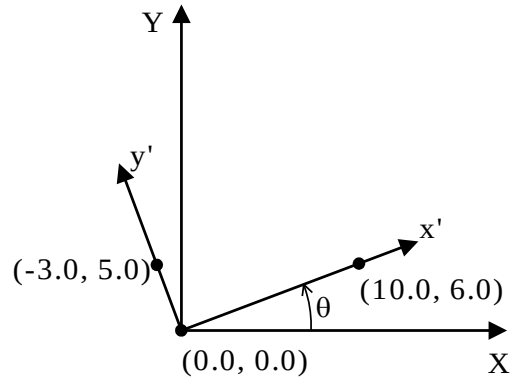
where {x´}, {y´} and {z´} are vectors defining the local coordinate directions (see figure below for a 2D example).

5. The Cartesian set transformation matrix will be checked for orthogonality and each vector will be automatically normalised.

6. Any number of CARTESIAN SET data chapters may be defined, but if duplicate **`nset`** values are specified the last definition will be used.



Definition of Cartesian Sets



$$[R]=\begin{bmatrix}10.0 & 6.0\\ -3.0 & 5.0\end{bmatrix} \quad \text{or} \quad [R]=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}$$

Cartesian Set Matrix Definition for 2D Example

Transformed Freedoms

Transformed freedoms may be used to rotate the global degrees of freedom at a node to a new orientation defined by the CARTESIAN SET command. The procedure is useful for applying loading values or boundary conditions in local coordinates directions.

The data section TRANSFORMED FREEDOMS is used to define the nodes to which a predefined local Cartesian coordinate set applies.

TRANSFORMED FREEDOMS [TITLE title]

N N_{last} N_{diff} nset

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical transformed freedoms.

nset The number of the Cartesian Set defining the directions or transformation of the transformed freedoms (see Cartesian Sets).

Notes

1. All concentrated loads, prescribed displacements, support conditions and constraint equations applied to a nodal freedom which has been transformed, act in the transformed directions.
2. Default output is in the global direction. Option 115 outputs values in the transformed directions.
3. The order of the transformed directions for cylindrical sets is defined as r´, t´, n´.

Constraint Equations

Nodal freedoms can be linked by a linear equation. Such constraint equations are useful in defining boundary conditions. For example, if an edge of a structure is to move as an integral unit, the appropriate translational degree of freedom at each node can be forced to act in a relative manner via constraint equations.

The data section CONSTRAINT EQUATIONS is used to define the equation coefficients and the nodes to which the specified constraint equation applies.

```
CONSTRAINT EQUATIONS  [TITLE  title]
EQUATION  C
N N_last N_diff  N_v  C_f
```

where:

C The constraint equation constant

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical variables and coefficients.

N_v The degree of freedom variable number at the node. For example, u=1, v=2 for plane stress elements (refer to *LUSAS Element Reference Manual*). See *Notes* on FREEDOM TEMPLATE.

C_f The coefficient corresponding to the node and variable number.

Notes

1. The general form of the equation is:

   C1V1+C2V2+ … +CiVi=C

   where Vi represents the variables and i the total number of coefficients
2. Each new equation starts with an EQUATION line followed by the corresponding constraint equation data lines.
3. If a node and variable number are repeated, the new coefficient overwrites the previous coefficient for that equation.
4. Constraint equations must not be over-sufficient for a unique solution. For example:

   U21 = U25, U25 = U30, U30 = U21 is over sufficient

   U21 = U25, U25 = U30 is sufficient
5. When using CONSTRAINT EQUATIONS with eigenvalue extraction, the constant C must be specified as zero.
6. If non-structural nodes or unconnected parts of a structure are to be constrained, the SYSTEM variable PENTLY should be specified to invoke the penalty constraint technique within the constraint equation. PENTLY should be specified as a small number to avoid numerical problems, but to ensure the reduction process within the solution does not break down.

7. Constrained variables which have been transformed will be constrained in their transformed directions.
8. If the FREEDOM TEMPLATE data chapter has been specified the values specified for Nv relate to the modified freedom list.
9. Freedoms featuring in constraint equations will not be candidates for use as automatic masters.
10. Care should be taken when specifying constraint equations so that ill conditioning of the equations does not result. For example, for some analyses it may be better to define:

    U21 = U25, U21 = U26, U21 = U27, U21 = U28

    rather than:

    U21 = U25, U25 = U26, U26 = U27, U27 = U28

11. If constraint equations are defined in a Guyan analysis or eigenvalue analysis, a Sturm sequence check cannot be carried out.
12. If constraint equations have been defined the Eigenvalue/Frequency Range facility cannot be used in an Eigenvalue analysis.
13. Constraint equations are not permissible for use with explicit dynamics elements.

Example. Constraint Equations

Plane stress problem with U displacements at nodes 9,11 and 15 constrained to be equal.

U9 = U11 U9 = U15

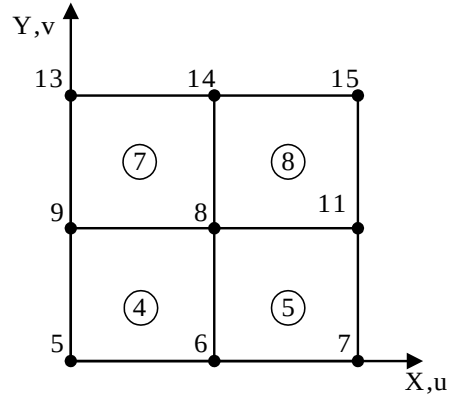
```
CONSTRAINT EQUATIONS
EQUATION 0.0
  9  0 0   1  1.0
  11 0 0   1 -1.0
EQUATION 0.0
  9  0 0   1  1.0
  15 0 0   1 -1.0
```

Plane stress problem with a general constraint relationship.

U9 + U11 + 0.5U6 - 3.0V14 = 12.5

```
CONSTRAINT EQUATIONS
EQUATION 12.5
```

```
9  11 2   1  1.0
6  0  0   1  0.5
14 0  0   2 -3.0
```



Support Conditions

The data section SUPPORT NODES is used to define the boundary conditions of finite element discretisation. Note that prescribed nodal displacements may also be defined using this data section.

SUPPORT NODES [TITLE title]

N N_{last} N_{diff} < $type_i$ >$_{i=1,n}$ [< V_i >$_{i=1,n}$]

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical supports.

$type_i$ The support type for each global freedom at a node.
=R for a restrained support freedom or a restrained support freedom with prescribed displacement.
=F for a free support freedom.
=S for spring freedom. (see *Note* on use of FREEDOM TEMPLATE)

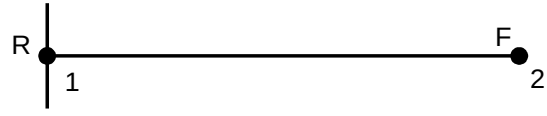
V_i The corresponding values of prescribed displacement or spring stiffness for each global freedom at a node. Rotational displacements and rotational spring stiffnesses should be prescribed in radians and stiffness/radian, respectively. (see *Note* on use of FREEDOM TEMPLATE)

n The MAXIMUM number of freedoms at a node for the structure being solved.

Notes

1. If a node number is repeated the new support (types and values) overwrites the previous support and an advisory message is printed out.

2. The freedom associated with a prescribed displacement **must** be restrained using this data section.
3. If support values **`Vi`** are specified, the total number of values must be equal to the MAXIMUM number of freedoms at **any** node of the structure. Support values corresponding to free freedoms must be given values, say **`Vi`**=0.
4. Ensure that the structure is restrained against translation and rotation in all global directions for all static analyses.
5. For a skew support use TRANSFORMED FREEDOMS, or a joint element with the appropriate orientation, spring stiffness (K=0 for free freedom) and initial strain (prescribed displacement).
6. Support nodes may only be modified between increments of a transient/dynamic nonlinear problem subject to the following conditions:
   - ❑ Only the support conditions for the nodes respecified will be modified.
   - ❑ A support condition may be respecified only for a node specified as a support in the first set of SUPPORT NODES. A dummy support may be specified i.e. all variables free, for nodes that are subsequently restrained.
7. Ensure that nodes are not free to rotate when attached to beam elements with free ends. For example, node 1 in the diagram below must be restrained against rotation as well as displacement otherwise the element will be free to rotate as a rigid body.



8. Support nodes may be omitted for eigenvalue analyses provided a shift is used in the EIGENVALUE CONTROL data chapter.

   If the FREEDOM TEMPLATE data chapter has been used care should be taken that the required support conditions relate to the modified freedom list.

   For nonlinear problems it is recommended that prescribed displacements are specified using the PDSP or TPDSP loading data chapters.

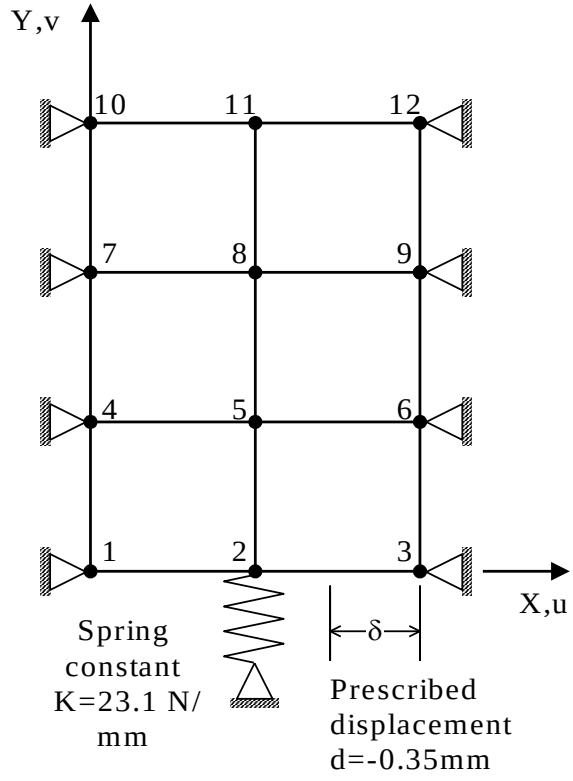
   The PDSP values supersede the values specified on the SUPPORT NODES data line. If a spring support is defined at a variable then any subsequent PDSP applied to that variable is read as the spring stiffness.
9. In a axisymmetric analysis the nodes on the axis of symmetry must be restrained to prevent translation across the syemmtry axis. i.e. For symmetry about Y displacement in x must be restrained.
10. For axisymmetric Fourier problems the restrictions shown below are applied to the freedoms of nodes lying on the axis of symmetry. These conditions are automatically imposed on the centre-line nodes.

axisymmetric about X axis		**axisymmetric about Y axis**	
harmonic	restraint	harmonic	restraint

n=0	v, w=0	n=0	u, w=0
n=1	u=0	n=1	v=0
n>1	u, v, w=0	n>1	u, v, w=0

Example 1. Support Conditions

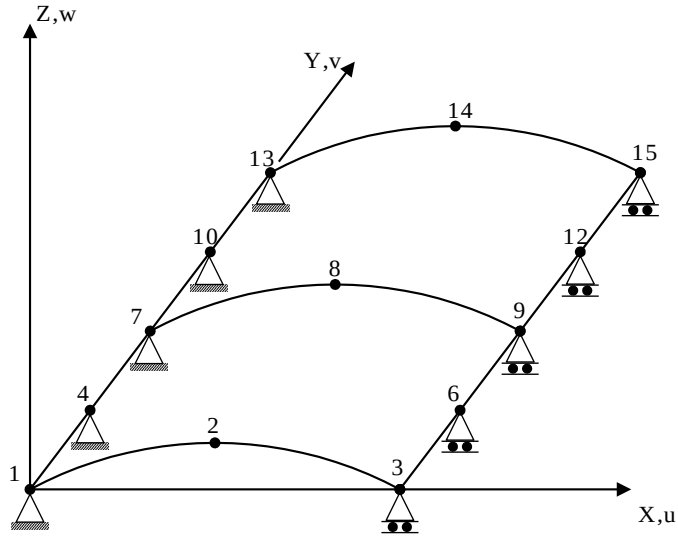
```
SUPPORT NODES
1 10  3  R R
2  0  0  F S   0.0   23.1
3 12  3  R F  -0.35   0.0
```



Example 2. Support Conditions

```
SUPPORT NODES
```

```
1 13 3 R R R F F
3 15 3 F R R F F
```



Coupled Analysis

In a thermo-mechanically coupled analysis the control of data to and from an external datafile generally requires three operations

- **Generation and initialisation of the Coupled Datafile**
- **Reading from the Coupled Analysis Datafile**
- **Writing to the Coupled Analysis Datafile**

For further information, refer to Coupled Analyses.

Initialisation of the Coupled Datafile

The data section COUPLE controls both the generation of the external coupled analysis datafile in the primary analysis and the file name specification in the secondary analysis

```
COUPLE  [OPEN]  [PARALLEL]  fname
```

OPEN Specifies that the analysis is to open the data transfer file. If omitted, the analysis will expect to find the data transfer file in its work area.

PARALLEL Specifies that a second analysis is running. If information is requested from the data transfer file and it is not located, an error is signalled if no other analysis is running, otherwise the current analysis will wait for new

data to become available. Note that the specification of **PARALLEL** without the second analysis running may cause the system to wait indefinitely for information that will not be available. The SYSTEM parameter **MXWAIT** determines the waiting time in seconds before an error is printed and the analysis terminated.

fname The name of data transfer file. The name must be less than 65 characters with a 3 letter extension, or 61 without an extension (an extension .DTF will automatically be added).

Reading from the Coupled Analysis Datafile

The data section COUPLE READ controls the reading from the coupled analysis datafile. The command is used to initialise the read condition at the start of an analysis (mandatory) and may, if required, be specified at later stages of the analysis.

```
COUPLE READ  [nstepr timerd ndatr nitemr]
```

nstepr The step number in the current analysis at which the first data read will occur. This parameter is to be used for static analyses (default = previous step [0], set to -1 to disable).

timerd The time in the current analysis at which data should be read. This parameter is to be used for dynamic/transient analyses (default = time end of previous step, set to -1 to disable).

ndatr The index number for the next dataset to be read (see *Notes*) (default = 1).

nitemr The type of data to be transferred
=1 for nodal coordinates
=2 for nodal temperatures
=3 to initialise nodal reference temperatures to those of the previous step (automatically switches to type 2 on subsequent steps).
=4 for heat flux due to plastic work and nodal coordinates
(default value: Structural analysis = 2, Thermal analysis = 1).

Notes

1. If the reference temperature of the structure is to be initialised with zero temperature **nitemr** type 3 should be used at the commencement of the analysis (the first specification of the COUPLE READ command).
2. Option 70 switches on the data echo. All data read into the analysis is echoed to the output file.
3. Each dataset that is written to the coupled datafile is given an integer index number which is automatically incremented by 1 on each write to the coupled analysis datafile. This number corresponds to **ndatr** specified in the COUPLE READ data line.
4. Further reading from the coupled datafile is controlled by the INCREMENTAL COUPLE READ data line within the respective analysis control data chapters.

Writing to the Coupled Analysis Datafile

The data section COUPLE WRITE controls writing to the coupled analysis datafile. The command is used to initialise the write conditions at the start of an analysis (mandatory) and may, if required, be specified at later stages of the analysis.

```
COUPLE WRITE  [nstepw timewt ndatw nitemw]
```

nstepw The step number in the current analysis at which the first data write will occur. This parameter is to be used for static analyses (default = current step [1], set to -1 to disable).

timewt The time in the current analysis at which data should be written. This parameter is to be used for dynamic/transient analyses (default = time at end of current step, set to -1 to disable).

ndatw The index number for the next dataset to be written (default = 1).

nitemw The type of data to be transferred
= 1 for nodal coordinates
= 2 for nodal temperatures
= 4 for heat flux due to plastic work and nodal coordinates
(default value: Structural analysis = 1, Thermal analysis = 2).

Notes

1. Each dataset that is written to the coupled datafile is given an integer index number which is automatically incremented by 1 on each write to the coupled analysis datafile. This number corresponds to **ndatw** specified in the COUPLE WRITE data line.
2. Further writing to the coupled datafile is controlled by the INCREMENTAL COUPLE WRITE data line within the respective analysis control data chapters.

Structural Loading

LUSAS incorporates a variety of loading types. The loading types available are classified into the following groups (the abbreviations for each loading type are shown in brackets):

- **Prescribed variables** (PDSP, TPDSP)
- **Concentrated loads** (CL)
- **Element loads** (ELDS, DLDL, DLDG, DLEL, DLEG, PLDL, PLDG)
- **Distributed loads** (UDL, FLD)
- **Body forces** (CBF, BFP, BFPE)
- **Velocities and accelerations** (VELOCITY, ACCELERATION)
- **Initial stresses and strains** (SSI, SSIE, SSIG)
- **Residual stresses** (SSR, SSRE, SSRG)
- **Temperature loads** (TEMP, TMPE)
- **Field loads** (ENVT, TDET, RIHG)

It is important for you to consult the individual element description in the *LUSAS Element Reference Manual*in order to check that the required loading is available for that particular element.

Two forms of nodal loading are possible. Firstly, a load may be applied at a node. This load will act on **all** elements which are connected to the node. Secondly, a load may be applied on an element node, where the load is applied at the node of the prescribed element only. If the required direction of a global load does not lie in the global axes then transformed freedoms can be used to transform the loads to the required local directions (this applies to CL, PDSP, TPDSP, VELOCITY and ACCELERATION loads only).

For some of the loading types it is possible to abbreviate a long line of data input. Such loading definitions make use of two parameters. The first, the loading data list pointer 'l', indicates the position in the loading list of the first required component. The second, 'n', indicates the total number of loading components which are required. For example, the full loading data list of temperature loading (TEMP) for an isoflex plate element contains the 8 components:

$$T \quad \frac{\partial T}{\partial x} \quad \frac{\partial T}{\partial y} \quad \frac{\partial T}{\partial z} \quad T_0 \quad \frac{\partial T_0}{\partial x} \quad \frac{\partial T_0}{\partial y} \quad \frac{\partial T_0}{\partial z}$$

where only $\frac{\partial T}{\partial z}$ and $\frac{\partial T_0}{\partial z}$ are applicable.

Hence, final and initial temperature gradients of 10.5 and 5.6 may be applied by specifying the full loading list (that is 8 required components) as:

```
LOAD CASE
TEMP 8
0 0 0 10.5 0 0 0 5.6
```

or, using the abbreviated loading input (using 5 components, and a position pointer of 4), as:

```
LOAD CASE
TEMP 5 4
10.5 0 0 0 5.6
```

Note. The default pointer position is 1 (that is, the start of the loading list). Note also that the values not required within the abbreviated loading list must be specified (as zeros).

Load Case Definition

The data section LOAD CASE is used to define the loading types which belong to a particular load case group. For linear analyses each specified LOAD CASE data section defines a new set of loading which are to be applied individually to the structure. For nonlinear analyses the LOAD CASE data section may also be used to define the incrementation of a set of applied loads.

LOAD CASE [TITLE title]

Notes

1. Each new load case must start with this header line.
2. Each LOAD CASE line must be followed by a load type line and corresponding load data lines; see subsequent sections.
3. Each LOAD CASE may consist of a number of load types.
4. Multiple linear LOAD CASES are processed simultaneously.

Prescribed Variables

Nodal variables can be prescribed in an incremental manner (PDSP) or in total form (TPDSP). In each case the variable to be prescribed must be restrained under the SUPPORT NODES data chapter.

Incremental Prescribed Variables (PDSP)

The data section PDSP is used to define incremental prescribed nodal variables.

PDSP n [l] [TITLE title]

N N_{last} N_{diff} < V_i > =1,n

n The required number of prescribed values.

l The starting location of the first input value in the prescribed displacement data list (default **l**=1)

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical prescribed values.

V_i The prescribed values (see *Note* on FREEDOM TEMPLATE).

Notes

1. The number of prescribed values must not exceed the number of freedoms for any node.
2. Prescribed values will function only if the corresponding freedoms are specified as restrained (R) in SUPPORT NODES.
3. If the FREEDOM TEMPLATE data chapter has been specified prescribed values will relate to the modified freedom list.

4. Prescribed nodal variables may also be defined via the SUPPORT NODES data chapter (this is not recommended for nonlinear analyses). In this case the prescribed velocities will apply to all loadcases.
5. Incremental and total prescribed displacements (see *Total prescribed variables (TPDSP)* below) should not be applied in the same analysis if load curves have been defined. It is recommended that total prescribed displacements are used with load curves.
6. Incremental and total prescribed displacements must not be combined to prescribe values for variables at the same node.
7. Rotational displacements should be specified in radians.
8. In a linear analysis, multiple PDSP load cases may be defined but there must be **no** change in the degrees of freedom that are loaded.

Total Prescribed Variables (TPDSP)

The data section TPDSP is used to define total prescribed nodal variables.

```
TPDSP  n  [l]  [TITLE  title]
N N_last N_diff  < V_i > =1,n
```

n The required number of prescribed values.

l The starting location of the first input value in the prescribed displacement data list (default **l**=1).

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical prescribed values.

V_i The prescribed values (see *Note* on FREEDOM TEMPLATE).

Notes

1. The number of prescribed values must not exceed the number of freedoms for any node.
2. Prescribed values will function only if the corresponding freedoms are specified as restrained (R) in SUPPORT NODES.
3. If the FREEDOM TEMPLATE data chapter has been specified prescribed values will relate to the modified freedom list.
4. Total and incremental prescribed displacements (see *Incremental prescribed variables (PDSP)* above) should not be applied in the same analysis if load curves have been defined. It is recommended that total prescribed displacements are used with load curves.
5. Total and incremental prescribed displacements must not be combined to prescribe values for variables at the same node.
6. Rotational displacements should be specified in radians.
7. In a linear analysis, multiple PDSP load cases may be defined but there must be **no** change in the degrees of freedom that are loaded.

Concentrated Loads (CL)

The data section CL is used to define concentrated forces and/or moments which are applied directly to structural nodes.

```
CL  [TITLE  title]
N N_last N_diff  < P_i >_i=1,n
```

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical values.

P_i The nodal forces/moments in global X, Y or Z directions (see *Notes*).

n The number of nodal forces/moments

Notes

1. If the FREEDOM TEMPLATE data chapter has been specified concentrated loading must relate to the modified freedom list.

Element Loads

Four types of internal element loads may be applied:

- **Element loads** (ELDS)
- **Distributed loads** (DLDL,DLDG)
- **Distributed element loads** (DLEL,DLEG)
- **Element point loads** (PLDL,PLDG)

Consult the relevant element section in the *LUSAS Element Reference Manual f*or details of the required loading components.

Element Loads (ELDS)

The data section ELDS is used to define the element load type and components.

```
ELDS  m  [l]  [TITLE  title]
L L_last L_diff  ltype  < V_i >_i=1,n
```

m Maximum number of element loads applied to any one element per load case.

l The starting location of the first input value in the element load data list (default **l**=1).

L L_{last} L_{diff} The first element, last element and difference between elements of the series of elements with identical element loading values.

ltype The element load type number. The element loading types available are:

11 Point loads and moments in local directions
12 Point loads and moments in global directions
21 Uniformly distributed loads in local directions
22 Uniformly distributed loads in global directions

23 Uniformly ditributed projected loads in global directions
31 Distributed element loads in local directions
32 Distributed element loads in global directions
33 Distributed element projected loads in global directions
41 Trapezoidal loads in local directions
42 Trapezoidal loads in global directions
43 Trapozoidal projected loads in global directions

V_i The element internal distances and load values.

n The number of element internal distances and load values for the loaded element type (see *Loading* section in the *LUSAS Element Reference Manual*).

Distributed Loads Local/Global (DLDL,DLDG)

The data sections DLDL and DLDG are used to define the distributed load components in local and global directions respectively.

{DLDL | DLDG} n [l] [TITLE title]

L N NL < q_i >$_{i=1,n}$

n The required number of values in the distributed load data list.

l The starting location of the first input value in the distributed load data list (default **l**=1).

L The element number.

N The element node number as defined in ELEMENT TOPOLOGY.

NL The distributed load sequential number (see *Notes*).

q_i The distributed load values.

Notes

1. Use the header line DLDL for local, or DLDG for global, distributed loads.
2. A unique sequential number must be assigned to each distributed load that is applied to an element for a given load case.
3. If an element has a sequential number that is repeated for a given load case, the new loads overwrite the previous values.

Distributed Element Loads Local/Global (DLEL,DLEG)

The data sections DLEL and DLEG are used to define the element distributed load components in local and global directions respectively.

{DLEL | DLEG} n [l] [TITLE title]

L L_{last} L_{diff} NL < q_i >$_{i=1,n}$

n The required number of values in the distributed internal element load data list.

l The starting location of the first input value in the distributed internal element load array (default **l**=1).

L L_{last} L_{diff} The first element, last element and difference between elements of the series of elements with identical distributed internal element load.

NL The distributed internal load number (see *Notes*).

q_i The distributed internal load values.

Notes

1. Use the header line DLEL for local, or DLEG for global, distributed internal element loads.
2. A unique internal load number must be assigned to each distributed load that is applied to an element for a given load case.
3. If an element has an internal load number that is repeated for a given load case, the new loads overwrite the previous values.

Element Point Loads Local/Global (PLDL,PLDG)

The data sections PLDL and PLDG are used to define the loading components for element point loads in local and global directions respectively.

{PLDL | PLDG} n [l] [TITLE title]

L L_{last} L_{diff} NL $< p_i >_{i=1,n}$

n The required number of values in the internal element point load data list.

l The starting location of the first input value in the internal element point load list (default **l**=1).

L L_{last} L_{diff} The first element, last element and difference between elements of the series of elements with identical internal element point loads.

NL The element internal point load sequential number.

P_i The element internal point load values.

Note

1. Use the header line PLDL for local, or PLDG for global, point loads.

Distributed Loads

Two types of distributed loads are available:

- **Uniformly Distributed Loads** (UDL)
- **Face Loads** (FLD)

Uniformly Distributed Loads (UDL)

The data section UDL is used to define the loading intensity components for uniformly distributed loads.

UDL [TITLE title]

L L_{last} L_{diff} < W_i >$_{i=1,n}$

L L_{last} L_{diff} The first element, last element and difference between elements of the series of elements with UDL loads.

W_i The uniformly distributed load (applied in the element local directions, see the *LUSAS Element Reference Manual*).

n The number of uniformly distributed load components.

Note

1. All values applied to elements are accumulative within each LOAD CASE.

Face Loads (FLD)

The data section FLD is used to define the loading components and associated element faces for face loading. The element face numbering conventions are shown in the *LUSAS Element Reference Manual*.

FLD n [l] [TITLE title]

L LF N < FL_i >$_{i=1,n}$

n The required number of values in the element face load data list.

l The starting location of the first input value in the element face load data list (default **l**=1).

L The element number.

LF The element face number (see the *LUSAS Element Reference Manual*).

N The element face node number as input in element topology.

FL_i The face load values.

Notes

1. If zero element face node **N**, is specified, then the face load will be applied to all nodes on the face.

2. When using Option 123 for clockwise node numbering, care should be taken to ensure that loading is applied in the correct direction.

Body Forces

Three types of body force loading are available:

- **Constant Body Forces** (CBF)
- **Body Force Potentials** (BFP)
- **Element Body Force Potentials** (BFPE)

Constant Body Forces (CBF)

The data section CBF is used to define the loading components for constant body force loading (specified as forces per unit volume).

CBF [n] [l] [TITLE title]

L L_{last} L_{diff} < q_i >$_{i=1,n}$

n The number of constant body forces/angular velocities/angular accelerations.

l The starting location of the first input value in the element load data list (default **l**=1).

L L_{last} L_{diff} The first element, last element and difference between elements of the series of elements with identical constant body forces.

q_i The constant body forces/angular velocities (see Notes)/angular accelerations in/about the global X, Y and Z directions/axes (see the *LUSAS Element Reference Manual*).

Notes

1. Option 48 switches the constant body force input to linear acceleration input. The angular velocities and accelerations remain unchanged.
2. All values applied to elements are accumulative within each LOAD CASE.
3. Option 102 switches off the load correction stiffness matrix due to centripetal acceleration.
4. Centripetal stiffening effects are limited to 2D-continuum, axisymmetric solid, 3D-continuum, semiloof shells, thick shells and 3D numerically integrated beam elements. They are only included in nonlinear analyses via the Total Lagrangian geometrically nonlinear facility.
5. Element loading will be a function of the square of any angular velocity specified. If auto incrementation is used in a nonlinear analysis the load factor, TLAMDA, will be applied to the equivalent element loading and not the angular velocity. If LUSAS Solver detects this combination an error message will be written to the output file and the analysis terminated. To override this error check OPTION 340 can be specified in the data file and the analysis will continue giving a warning message only. To directly control the magnitude of the angular velocity applied to the structure in a nonlinear analysis manual incrementation or load curves should be utilised.

Body Force Potentials (BFP)

The data section BFP is used to define the loading components for body force potential loading (specified as forces at nodes).

BFP [TITLE title]

N N_{last} N_{diff} < q_i >$_{i=1,n}$

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with force potentials.

q_i The body force potentials/pore water pressure/constant body forces at nodes in global and/or local directions (see *Notes*).

n The number of body force potentials/pore water pressure/constant body forces.

Notes

1. BFP values at a node apply to all elements connected to that node.
2. The *LUSAS Element Reference Manual* must be consulted to find out which values can be specified under BFP loading for a particular element. Unless otherwise stated, the values are defined as force/unit area or volume and consist of **qi** where:
   - **i**=1 to 3 body force potentials, applied in local element directions
   - **i**=4 pore water pressure, applied in global directions
   - **i**=5 to 7 constant body forces, applied in global directions
3. All values applied to elements are accumulative within each LOAD CASE.

Element Body Force Potentials (BFPE)

The data section BFPE is used to define the loading components for elemental body force potential loading (specified as forces at element nodes).

BFPE n [l] [TITLE title]

L N $< \Phi_i >_{i=1,n}$

n The required number of values in the body force potential data list.

l The starting location of the first input value in the body force potential data list (default **l**=1).

L The element number.

N The element node number as defined in ELEMENT TOPOLOGY.

Φ_i The body force potential values at the element node in global and/or local directions (see *Notes*).

Notes

1. The *LUSAS Element Reference Manual* must be consulted to find out which values can be specified under BFPE loading for a particular element. Unless otherwise stated, the values are defined as force/unit area or volume and consist of:
   - Φi:
     - **i**=1 to 3 body force potentials, applied in local element directions
     - **i**=4 pore water pressure, applied in global directions
     - **i**=5 to 7 constant body forces, applied in global directions
2. All values applied to elements are accumulative within each LOAD CASE.

Velocity and Acceleration

In dynamic analyses, velocities or accelerations at a nodal variable can be defined. These values can be used to specify an initial starting condition or they may be prescribed for the whole analysis. If values are to be prescribed throughout the analysis load curves must be used (see Curve Definition) and the appropriate freedom must be restrained in the SUPPORT NODES data chapter.

- **Prescribed Velocities (VELOCITY)**
- **Prescribed Accelerations (ACCELERATION)**
- **Examples of Acceleration and Velocity Curves**

Prescribed Velocities (VELOCITY)

The data section VELOCITY may be used to define the components of an initial velocity or to prescribe velocities throughout a dynamic analysis.

VELOCITY n [l] [TITLE title]

N N_{last} N_{diff} < V_i >$_{i=1,n}$

n The required number of velocity values in the data list.

l The starting location of the first input value in the velocity data list (default **l**=1).

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical velocities.

V_i The velocity components (see *Notes*).

Notes

1. If a component of velocity is to be prescribed throughout an analysis the appropriate freedom must be specified as restrained (R) in support nodes. If an initial velocity is defined without using load curves the support condition for the variable must be free.
2. The number of velocities must not exceed the number of freedoms for any node.
3. Prescribed or initial velocities are only applicable to dynamic analyses.
4. Initial velocities should only be applied to the first load case (i.e. at time=0).
5. If the FREEDOM TEMPLATE data chapter has been specified velocities will relate to the modified freedom list.
6. In general, load curves (see Curve Definition) must be used to prescribe velocities in an analysis. However, initial velocities may be defined without using load curves if no other load type is controlled by a load curve.
7. In general, it is not reasonable to prescribe velocities and accelerations (see ACCELERATION below) for the same variable at the same point in time; if this does occur in an analysis the acceleration will overwrite the velocity and a warning message will be output. An exception to this rule occurs for implicit dynamics

analyses where an initial velocity and acceleration may be used to define an initial condition for the same variable.

8. If initial conditions are to be applied, refer to Transient Dynamic Analysis for details on how to compute the data input required for the appropriate dynamics integration scheme.
9. In explicit dynamics, if an initial velocity is to be followed by subsequent prescribed values (velocities, accelerations or displacements) at the same variable, the appropriate freedom must be restrained from the outset.
10. Velocities defined in load curves for explicit dynamics will be written to the output file one time step later than the time at which they were defined. This time lag is a consequence of the central difference integration scheme.
11. It is possible to switch from a prescribed velocity to a prescribed acceleration (or vice-versa) for the same variable by manipulating load curve data. Care should be taken when doing this to avoid any discontinuity which could excite a high frequency response in the model. An example of manipulating load curve data in this way is given on Examples of Acceleration and Velocity Curves.

Prescribed Accelerations (ACCELERATION)

The data section ACCELERATION may be used to define the components of an initial acceleration or to prescribe accelerations throughout a dynamic analysis. It should be noted however that an initial acceleration cannot be used to define the starting conditions for an explicit dynamic analysis.

ACCELERATION n [l] [TITLE title]

N N_{last} N_{diff} < V_i >$_{i=1,n}$

n The required number of acceleration values in the data list.

l The starting location of the first input value in the acceleration data list (default **l**=1).

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical accelerations.

V_i The acceleration components (see *Notes*).

Notes

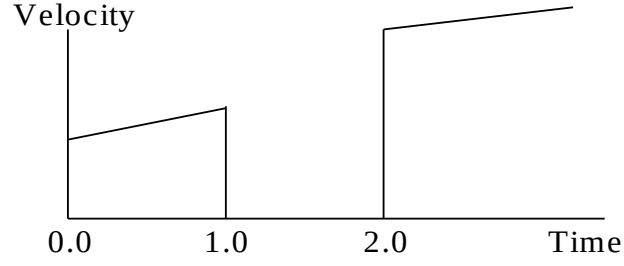
1. If a component of acceleration is to be prescribed throughout an analysis the appropriate freedom must be specified as restrained (R) in support nodes. If an initial acceleration is defined in an implicit dynamic analysis without using load curves the support condition for the variable can be free or restrained.
2. The number of accelerations must not exceed the number of freedoms for any node.
3. Prescribed or initial accelerations are only applicable to dynamic analyses.
4. Initial accelerations should only be applied to the first load case (i.e. at time=0) and are only valid for implicit dynamic analyses.

5. If the FREEDOM TEMPLATE data chapter has been specified accelerations will relate to the modified freedom list.
6. In general, load curves (see Curve Definition) must be used to prescribe accelerations in an analysis. However, initial accelerations may be defined without using load curves in an implicit dynamic analysis if no other load type is controlled by a load curve.
7. In general, it is not reasonable to prescribe accelerations and velocities (see VELOCITY above) for the same variable at the same point in time; if this does occur in an analysis the acceleration will overwrite the velocity and a warning message will be output. An exception to this rule occurs for implicit dynamics analyses where an initial velocity and acceleration may be used to define an initial condition for the same variable.
8. If initial conditions are to be applied, refer to Transient Dynamic Analysis for details on how to compute the data input required for the appropriate dynamics integration scheme.
9. In explicit dynamics, accelerations defined in load curves will be written to the output file one time step later than the time at which they were defined. This time lag is a consequence of using the central difference integration scheme.
10. It is possible to switch from a prescribed acceleration to a prescribed velocity (or vice-versa) for the same variable by manipulating load curve data. Care should be taken when doing this to avoid any discontinuity which could excite a high frequency response in the model. An example of manipulating load curve data in this way is given below.

Examples of Acceleration and Velocity Curves

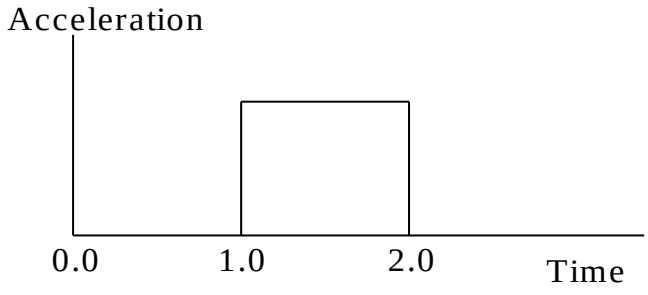
This example shows two methods for defining curves which alternate the prescribing of velocities and accelerations **at the same freedom**. In the first method, all load curves are assigned from time step 0 and zero sections in a curve allow control to be passed from one curve to another. In the second method, curves are reassigned at appropriate points in the analysis. Either method can be used in implicit dynamics analyses but only the first method should be used for explicit dynamics.

Example. Load Curves Method 1



Load Curve 1

Prescribe velocity between times 0.0 and 1.0 and from time 2.0 onwards



Load Curve 2

Prescribe acceleration between times 1.0 and 2.0

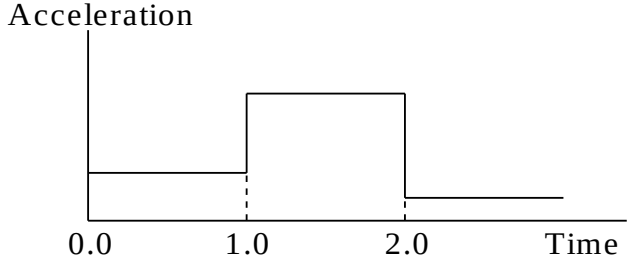
Loading section from data file:

```
LOAD CASE
 VELOCITY 2
 1  2  1  0.0  200.0
LOAD CASE
 ACCELERATION  2
 1  2  1  0.0  300.0
CURVE DEFINITION 1 USER
 0.0      1.0
 1.0      1.5
 1.00001  0.0
 1.99999  0.0
 2.0      3.0
 10.0     5.0
CURVE DEFINITION 2 USER
 0.0      0.0
 0.99999  0.0
 1.0      1.0
 2.0      1.0
 2.00001  0.0
 10.0     0.0
CURVE ASSIGNMENT
 1  1.0  1
 2  1.0  2
```

```
DYNAMIC CONTROL
 INCREMENTATION 0.01
 CONSTANTS             D
 OUTPUT          1
 TERMINATION           30
END
```

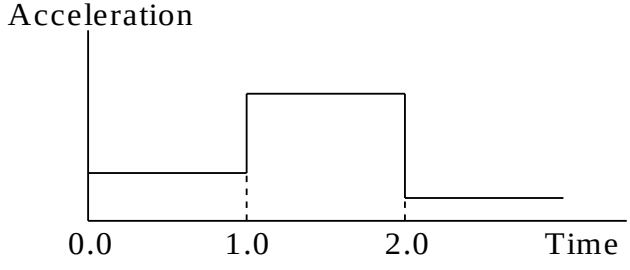
Combining curves with zero sections in this manner is only required if it is necessary to alternate the specification of velocities and accelerations at the same freedom. In explicit dynamics (with a varying time step size) the results near discontinuity points in the load curves should be checked to ensure that the tolerance used for defining zero sections is adequate enough to prevent erroneous values being prescribed. For implicit analyses the time step size is fixed and this potential problem will not arise if the load curves are defined correctly.

Example. Load Curves Method 2



Load Curve 1

Prescribe velocity profile completely



Load Curve 2

Prescribe acceleration profile completely

Note. Method 2 should only be used for implicit dynamic analyses.

Loading section from data file:

```
LOAD CASE
```

```
 VELOCITY 2
 1  2  1  0.0 100.0
LOAD CASE
 ACCELERATION   2
 1  2  1  0.0   100.0
CURVE DEFINITION 1 USER
 0.0      2.0
 1.0      3.0
 2.0      6.0
 10.0     10.0
CURVE DEFINITION 2 USER
 0.0      1.0
 0.99999  1.0
 1.0      3.0
 2.0      3.0
 2.00001  0.5
 10.0     0.5
CURVE ASSIGNMENT
 1  1.0  1
DYNAMIC CONTROL
 INCREMENTATION 0.01
 CONSTANTS            D
 OUTPUT         1
 TERMINATION          10
CURVE ASSIGNMENT
 2  1.0  2
DYNAMIC CONTROL
 TERMINATION          10
CURVE ASSIGNMENT
 1  1.0  1
DYNAMIC CONTROL
 TERMINATION          10
```

```
END
```

Initial Stresses and Strains

Three types of initial stress and strain loading are available:

- **Initial stresses and strains at nodes (SSI)**
- **Initial stresses/strains for elements (SSIE)**
- **Initial stresses and strains at gauss points (SSIG)**

Initial stresses and strains are applied as the first load case and subsequently included into the incremental solution scheme for nonlinear problems.

Initial stresses and strains are only applicable to numerically integrated elements. Refer to the individual element descriptions in the *LUSAS Element Reference Manual f*or details of the initial stress and strain components.

Initial Stresses and Strains at Nodes (SSI)

The data section SSI is used to define the initial stress and strain components at nodes.

```
SSI  n  [l]  [TITLE  title]
N N_last N_diff  < V_i >_i=1,n
```

n The required number of initial stresses or strains at a node.

l The starting location of the first input value in the element stress/strain data list (default, **l**=1 for stress input). The value **l**=**ndse** + 1 gives the starting location for strain input, where **ndse** is the number of stress components for the loaded element type.

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical initial stresses/strains.

V_i The initial stress, stress resultant or initial strain values at a node, relative to the reference axis.

Note

1. The initial stress/strain values at a node, apply to all elements connected to that node.

Example. Initial Stress

To apply an initial stress resultant (Mxy) of magnitude 2 to node number 10 of a QSI4 element would require the following command:

```
SSI  6
 10  0  0     0  0  0  0  0  2
```

and for a strain (yxy) at the same node of magnitude 0.001

```
SSI  6
 10  0  0     0  0  0  0  0  0    0  0  0 0  0  0.0001
```

or

```
SSI  6  7
 10  0  0 0  0  0  0  0  0.0001
```

Initial Stresses/Strains for Elements (SSIE)

The data section SSIE is used to define the initial element stress and strain components.

```
SSIE  n  [l]  [TITLE  title]
L  N  < Vi >i=1,n
```

n The required number of initial stresses or strains at an element node.

l The starting location of the first input value in the element stress/strain data list (default, **l**=1 for stress input). The value **l**=**ndse** + 1 gives the starting location for strain input, where **ndse** is the number of stress components for the loaded element type.

L The element number.

N The element node number as defined in ELEMENT TOPOLOGY.

V_i The initial stress, stress resultant or strain values at the element node.

Example. Initial Stress

To apply an initial stress (sy) of magnitude 2 to node number 7 of QPM8 element number 3 would require the following command:

```
SSIE  3
 3  7     0  2  0
```

and for a strain (ey) at the same node of magnitude 0.001

```
SSIE  3
 3  7     0  0  0    0  0.0001  0
```

or

```
SSIE  3  4
 3  7     0  0.0001  0
```

Initial Stresses and Strains at Gauss Points (SSIG)

The data section SSIG is used to define the initial stress and strain components at Gauss points.

SSIG n [l] [TITLE title]

L L_{last} L_{diff} NGP < V_i >$_{i=1,n}$

n The required number of initial stresses or strains at an element Gauss point.

l The starting location of the first input value in the element stress/strain data list (default, **l**=1 for stress input). The value **l**=**ndse** + 1 gives the starting location for strain input, where **ndse** is the number of stress components for the loaded element type.

L L_{last} L_{diff} The first element, last element and difference between elements of a series of elements with identical initial stress/strains.

NGP The Gauss point number.

V_i The initial stress, stress resultant or strain value at the element Gauss point relative to the reference axis.

Example. Initial Stress

To apply an initial stress (sxy) of magnitude 2 to Gauss point number 2 of QAX4 element number 10 would require the following command:

```
SSIG  4
 10  0  0    2    0  0  2  0
```

and for a strain (exy) at the same node of magnitude 0.001

```
SSIG  4
 10  0  0    2    0  0  0  0    0  0 0.0001  0
```

or

```
SSIG  4  5
 10  0  0 2    0  0  0.0001  0
```

Residual Stresses

Three types of residual stress loading are available:

- **Residual stresses at nodes (SSR)**
- **Residual stresses for elements (SSRE)**
- **Residual stresses at gauss points (SSRG)**

Residual stresses (unlike initial stresses) are assumed to be in equilibrium with the undeformed geometry and are not treated as a load case as such. They are considered as a starting position for stress for a nonlinear analysis. Failure to ensure that the residual stresses are in equilibrium will result in an incorrect solution.

Refer to the individual element descriptions in the *LUSAS Element Reference Manual f*or details of the residual stress components.

Residual Stresses at Nodes (SSR)

The data section SSR is used to define the components of residual stress at nodes.

```
SSR  n  [l]  [TITLE  title]
```

```
N N_last N_diff  < σ_ri >_i=1,n
```

n The required number of residual stresses at a node.

l The starting location of the first input value in the element residual stress data list (default **l**=1).

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical residual stresses.

σ_{ri} The residual stress or stress resultant values at a node, relative to the reference axis.

Note

1. The residual stress value at a node applies to all elements connected to that node.

Example. Residual Stress

To apply a residual stress resultant (Mxy) of magnitude 2 to node number 10 of a QSI4 element would require the following command:

```
SSR  6
 10  0  0     0  0  0  0  0  2
```

Residual Stresses for Elements (SSRE)

The data section SSRE is used to define the element residual stress components.

```
SSRE  n  [l]  [TITLE  title]
```

```
L  N  < σ_ri >_i=1,n
```

n The required number of residual stresses at an element node.

l The starting location of the first value in the element residual stress data list (default **l**=1).

L The element number.

N The element node number as defined in ELEMENT TOPOLOGY.

σ_{ri} The residual stress or stress resultant values at the element node, relative to the element reference axis.

Example. Residual Stress

To apply a residual stress (sy) of magnitude 2 to node number 7 of QPM8 element number 3 would require the following command:

```
SSRE  3
 3  7     0  2  0
```

Residual Stresses at Gauss Points (SSRG)

The data section SSRG is used to define the components of residual stress at Gauss points.

```
SSRG  n  [l]  [TITLE  title]
```

$\mathbf{L\ L_{last}\ L_{diff}\ \ NGP\ \ <\ \sigma_{ri}\ >_{i=1,n}}$

n The required number of residual stresses at an element Gauss point.

l The starting location of the first input value in the element residual stress data list (default **l**=1).

$\mathbf{L\ L_{last}\ L_{diff}}$ The first element, last element and difference between elements of the series of elements with identical residual stresses.

NGP The Gauss point number.

σ_{ri} The residual stress or stress resultant values for the element, relative to the element reference axis.

Example. Residual Stress

To apply a residual stress (sxy) of magnitude 2 to Gauss point number 2 of QAX4 element number 10 would require the following command:

```
SSIG  4
 10  0  0    2    0  0  2  0
```

Temperature Loads

Two types of temperature loading are available:

- **Temperature loads at nodes (TEMP)**
- **Temperature loads for elements (TMPE)**

Consult the individual element descriptions in the *LUSAS Element Reference Manual f*or details of the temperature components.

Temperature Loads at Nodes (TEMP)

The data section TEMP is used to define the loading components for nodal temperature loads.

```
TEMP  n  [l]  [TITLE  title]
```

$$\mathbf{N}\ \mathbf{N}_{last}\ \mathbf{N}_{diff}\ <\ \mathbf{T}_i\ >_{i=1,n}$$

n The required number of values in the temperature data list.

l The starting location of the first input value in the temperature data list (default **l**=1).

$\mathbf{N}\ \mathbf{N}_{last}\ \mathbf{N}_{diff}$ The first node, last node and difference between nodes of the series of nodes with identical temperature values.

$\mathbf{T}_i$ The temperature values at a node.

Notes

1. The temperature values at a node apply to all elements connected to that node, except joints, in which temperature loading is invoked using Option 119.
2. For step by step problems, the initial temperature values need only be specified on the first load step.
3. The TEMP data section may be used to provide a temperature field for computing initial material properties in a nonlinear analysis. To initialise the temperature field in a nonlinear field analysis, the temperature loading must be applied using a manual loading increment.
4. In a stress analysis, temperature loading will only induce stresses if the coefficient of thermal expansion is specified in the material properties.
5. To initialise the temperature field in a nonlinear field analysis, the temperature loading must be applied using a manual load increment.

Temperature Loads for Elements (TMPE)

The data section TMPE is used to define the loading components for element temperature loads.

```
TMPE  n  [l]  [TITLE  title]
```

$$\mathbf{L}\ \ \mathbf{N}\ \ <\ \mathbf{T}_i\ >_{i=1,n}$$

n The required number of values in the temperature data list.

l The starting location of the first input value in the temperature data list (default **l**=1).

L The element number.

N The element node number as defined in ELEMENT TOPOLOGY.

Ti The temperature values at the element node.

Notes

1. Temperature is only applied to the node of the element specified.
2. For step by step problems, the temperature values need only be specified on the first load step.
3. The TMPE data section may be used to provide a temperature field for computing initial material properties in a nonlinear analysis. To initialise the temperature field in a nonlinear field analysis, the temperature loading must be applied using a manual loading increment.
4. In a stress analysis, temperature loading will only induce stresses if the coefficient of thermal expansion is specified in the material properties.
5. To initialise the temperature field in a nonlinear field analysis, the temperature loading must be applied using a manual load increment.

General Point/Patch Loads

Since loading may not always conveniently be applied directly to the finite element mesh, the general point and patch load facilities provide a means of calculating the equivalent loads which are applied at the nodes to model loading which may lie within a single element or straddle several elements. By rotation of the vertical loading vector, in-plane loading may also be modelled.

General loading can be applied across the finite element mesh by the definition of a **search area**. The search area overlies the finite element mesh and is composed of equivalent 3 and 4 noded, or 6 and 8 noded triangular and quadrilateral elements, connected into an area which conveniently covers all or part of the structure to be loaded. The loading is applied to this overlying mesh and converted into equivalent nodal loads using the shape functions of the search elements. These nodal loads are then applied directly to the underlying structural mesh. Note that the nodal loads correctly represent the vertical and in-plane components but do not account for any equivalent bending moments.

To ensure a correct transfer of loading, the search area node numbers must coincide with the structural mesh. It is important to note that the loads are only applied to the nodes which are common to both the mesh and the search area.

The definition of general point and patch loads therefore generally involves four data sections:

- **Search area definition**
- **General point load definition**
- **General patch load definition**
- **Assigning point or patch loads to the structure**

An example showing the usage of these data sections is given on General Point and Patch Loading Assignment.

Search Area Definition

The data section SEARCH AREA is used to define the general point and patch loading search area.

```
SEARCH AREA  isarea
nelt  < node_i >_i=1,n
```

isarea The search area reference number.
nelt The element number allocated to the particular search element being defined.
nodei Node numbers for each node of the particular search element being defined.
n The number of nodes required to define the search element.

Notes

1. Each search element must be given a unique identifying number. The element numbers may, but not necessarily, be the same as those used in the ELEMENT TOPOLOGY.
2. Node numbers must be the same as those used for the structural mesh (defined in NODE COORDINATES).
3. The search element numbers may have omissions in the sequence and need not start at one. The order in which the elements are specified is arbitrary.
4. Nodes must be numbered in an anti-clockwise manner.
5. Linear three and four noded elements may be combined or similarly six and eight noded elements may be combined to mesh the search area.
6. A search area must form a continuous surface irrespective of the element types used to mesh the structure.
7. A search area may be defined on any face of an element.

General Point Load Definition

The data section POINT DEFINITION is used to define a series of point loads of arbitrary magnitude in a local z-coordinate system. The loading is applied to the structure by using the ASSIGN command. A CARTESIAN SET is used to define the local xy coordinate system in which the position of the load is defined; the local axes for the load are also assigned using the ASSIGN command.

```
POINT DEFINITION  ldefn
x y  P
```

ldefn The load reference number.
x The local x-coordinate of point load.
y The local y-coordinate of point load.
P The magnitude of the point load acting in the local z-direction.

Notes

1. The x and y coordinates are defined in local coordinates with a user defined origin and orientation (see ASSIGN data chapter)
2. A load may include any number of point loads which are allowed to be defined in an arbitrary order.
3. A positive **P** is applied in the positive local z direction.

General Patch Load Definition

The data section PATCH DEFINITION is used to define general patch loads. These are converted to a series of point loads which are then treated in the same fashion as the point loading. The loading is applied to the structure by the specification of the node of the origin of the local xy coordinate system and its orientation with respect to the global axes using the ASSIGN command. Each different load can be applied at a different point and orientation, where the orientation of the local axes is defined using the CARTESIAN SET command.

```
PATCH DEFINITION  ldefn ndivx ndivy
x y  P
```

ldefn	The patch reference number
ndivx	The number of patch divisions in local x-direction
ndivy	The number of patch divisions in local y-direction
x	The local x-coordinate of point
y	The local y-coordinate of point
P	The magnitude of the patch load acting in the local z-direction (see *Notes*).

Notes

1. The type of patch is defined by the number of points:
   - points = Straight line knife edge load
   - points = quadrilateral straight boundary patch
   - 8 points = curved boundary patch
2. For knife edge patch loads, the direction of **P** corresponds with the local z direction (i.e. a positive **P** is applied in the positive local z direction). For 4 and 8 point patch loads the direction of point ordering for the patch is important. If the patch points are defined in an anticlockwise direction (when looking from positive to negative local z) the direction of loading corresponds with the positive local z direction. If the patch points are defined in a clockwise direction, a positive **P** will be applied in the negative local z direction.
3. The angle subtended at any corner of a quadrilateral zone must be less than 180° otherwise non-uniqueness of mapping may result.

4. For the eight noded patch the mid-nodes must lie inside the central half of the sides.
5. The PATCH load is transformed to an equivalent summation of **ndivx*****ndivy** point loads; the more divisions used the more realistic the patch load.
6. For line loads the **ndivx** parameter defines the number of load divisions used.

General Point and Patch Loading Assignment

Defined general point and patch loading is applied to the structure via the ASSIGN data section. This incorporates a transformation which allows the loads, defined in local coordinates, to be orientated at any angle to the global axes. Firstly, the local point or patch loading is orientated with respect to the main structure. The vertical loading can then be rotated to give an in-plane loading component (for example, to represent a vehicle braking or cornering load).

ASSIGN

norg ldfen nset isarea lset factor

norg The node number of the origin of the point or patch loading local coordinate system.

ldfen The load definition number (see *Point Definition* and *Patch Definition* above).

nset The Cartesian set defining the orientation of the local point or patch loading coordinates to the global system (see Cartesian Sets). By default the load will be applied in global directions.

isarea The search area definition number (see Search Area Definition).

lset The Cartesian set defining the orientation of the applied vertical loading with respect to the global axes (see Cartesian Sets).

factor The factor to be applied to the vertical load (default = 1.0).

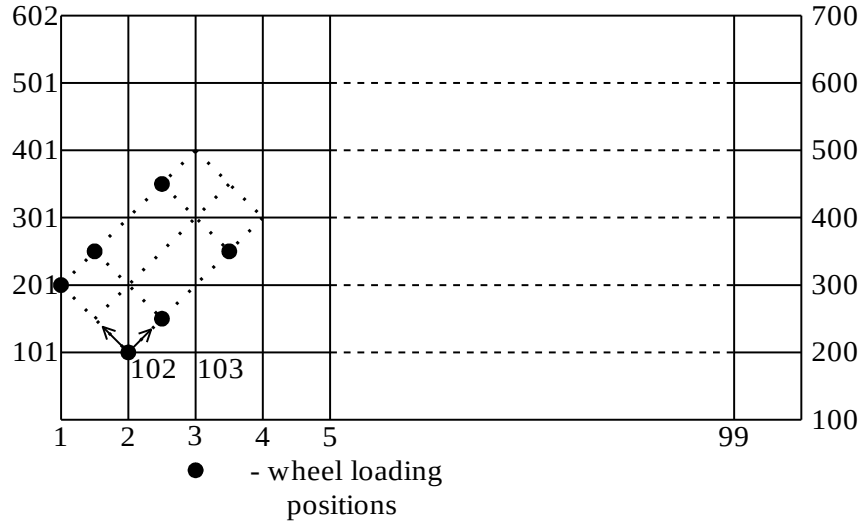
Notes

1. Any number of point or patch loads may be applied in an arbitrary order.
2. The origin of the local axes for point or patch loads may be placed on any node defined in the NODE COORDINATE data chapter. This node may be defined outside the boundary of the structure.
3. The orientation of point or patch loads may be varied using different Cartesian sets - (**nset** varies the patch orientation, **lset** varies the vertical load orientation). If zero is specified for the Cartesian set number, the orientation of point or patch loads is assumed to be with respect to the global axes.
4. Loading applied outside of the search area are ignored.
5. The translation degrees of freedom of the structural elements must agree with the resolved loading applied to the nodes; if the structure has, for instance, only a lateral freedom w, it is not possible to apply in-plane loading in the u and v

direction which arise if the lateral load applied to the loading mesh is reoriented about either the x or y axes.

Example. Lorry Loading on Bridge Deck

The bridge deck is discretised using a regular mesh of square planar elements:



1. Definition of orientation of local coordinate.

```
CARTESIAN SET
 1     2   103  102
```

2. Definition of a search area 2 using 4 noded elements.

```
SEARCH AREA 2
 FIRST   1    1    2  102  101
   INC   1    1    1    1    1   (5)
   INC   4  100  100  100  100   (5)
```

3. Definition of point loads to represent a 6 wheeled lorry load (local origin at node 102) with lorry at 45° to the global axes.

```
POINT DEFINITION 4
 0 0 3
 1 0 3
 3 0 2
```

```
0 2 3
1 2 3
3 2 2
```

4. Assign point loads to structure.

```
LOAD CASE
 ASSIGN
 102   4   1   2 0   1
```

Field Loading

Three types of loading are available for field analyses:

- **Environmental temperature loading (ENVT)**
- **Temperature dependent environmental temperature loading (TDET)**
- **Temperature dependent rate of internal heat generation (RIHG)**

The prescription of temperature dependent or radiation loading turns a linear field problem into a nonlinear field problem. Note that other loading types such as face loads, constant body force, body force potentials, etc. are also applicable to field problems.

Related commands are:

- Temperature load case
- Temperature Load Assignments

Environmental temperature loading (ENVT)

The data section ENVT is used to define the environmental temperature and convective and radiative heat transfer coefficients for environmental temperature loading.

```
ENVT  n  [l]  [TITLE  title]
L  LF  N  < V_i >_i=1,n
```

n The required number of values in the element environmental temperature data list (2 or 3).

l The starting location of the first input value in the element environmental temperature data list (default **l**=1).

L The element number.

LF The element face number.

N The element face node number as defined in ELEMENT TOPOLOGY.

V1 The environmental temperature.

V2 The convective heat transfer coefficient.

V3 The radiation heat transfer coefficient.

Notes

1. If heat transfer coefficients vary on a specified face the values will be interpolated using the shape functions to the Gauss points.
2. If a zero element face node number is specified, then the environmental load will be applied to all nodes on the face.
3. If a nonzero radiation heat transfer coefficient is specified, the problem is nonlinear and NONLINEAR CONTROL must be used.
4. When a radiation heat transfer coefficient is specified the temperature units for the problem will be Kelvin by default. Option 242 allows temperatures to be input and output in Celsius (Centigrade) for problems involving radiative heat transfer.
5. Load curves can be used to maintain or increment the environmental temperature as a nonlinear analysis progresses.
6. Automatic load incrementation within the NONLINEAR CONTROL data chapter can be used to increment ENVT loading.
7. When using load curves or auto incrementation with ENVT loading, the envionmental temperature may be incremented but the heat coefficients remain constant. This means that the heat coefficients will be applied even if the load curve defines a zero environmental temperature. If the heat coefficients and ENVT load are to be introduced or removed during an analysis, then the first (or last) point defining the load curve must coincide with the time at which the ENVT is to be introduced (or removed). Note that if the ENVT is inactive, the heat coefficients are not applied.
8. If the boundary heat transfer conditions can be adequately represented by non-temperature dependent convection and radiation heat transfer coefficients, the ENVT command may be used to input the data. ENVT loading modifies the stiffness matrix and, consequently, for linear field problems, only one load case may be solved in any one analysis when this loading is applied.

Temperature load case

The TEMPERATURE LOAD CASE data section is used to define a new temperature dependent (that is, nonlinear) temperature load case.

```
TEMPERATURE LOAD CASE  [TITLE  title]
```

Notes

1. The TEMPERATURE LOAD CASE data section must be issued if TDET or RIHG loading are required.
2. Each TEMPERATURE LOAD CASE definition must be directly followed by either TDET, RIHG and/or TEMPERATURE LOAD ASSIGNMENTS data sections.
3. Each load case may consist of any number of load types.

4. NONLINEAR CONTROL must be specified if temperature dependent loading is used.
5. The order of data input is first to list the temperature dependent data in the form of a table using the TEMPERATURE LOAD CASE followed by subcommands TDET and/or RIHG. These tables are then assigned to the elements using the TEMPERATURE LOAD ASSIGNMENTS followed by the subcommands TDET and/or RIHG.
6. If TDET or RIHG loading is to be combined with other load types such as CL or PDSP, the TEMPERATURE LOAD CASE must precede the LOAD CASE data chapter.

Temperature dependent environmental temperature loading (TDET)

The data section TDET is used to define the environmental temperature, convective and radiative heat transfer coefficients, and reference temperature for temperature dependent environmental temperature loading.

```
TDET  n  [l]  [TITLE  title]
TABLE  ilod
< Vi >i=1,n T
```

n The required number of values in the environmental temperature data list (must currently =3).

l The starting location of the first input value in the environmental data list (default **l**=1).

ilod The table reference number.

Vi The list of values:
- **V1** - Environmental temperature
- **V2** - Convective heat transfer coefficient
- **V3** - Radiative heat transfer coefficient

T Reference temperature.

Notes

1. When a radiation heat transfer coefficient is specified the temperature units for the problem will be Kelvin by default. Option 242 allows temperatures to be input and output in Celsius (Centigrade) for problems involving radiative heat transfer.
2. Load curves can be used to maintain or increment the environmental temperature as a nonlinear analysis progresses.
3. When using load curves incrementation with TDET loading, the environmental temperature may be incremented but the heat coefficients remain constant. This means that the heat coefficients will be applied even if the load curve defines a zero environmental temperature. If the heat coefficients and TDET load are to be introduced during an analysis, then the first (or last) point defining the load curve

must coincide with the time at which the ENVT is to be introduced (or removed). Note that if the ENVT is inactive, the heat coefficients are not applied.

4. Automatic load incrementation under the NONLINEAR CONTROL data chapter cannot be used with TDET loading.

Temperature dependent rate of internal heat generation (RIHG)

The data section RIHG is used to define the rate of internal heat generation and reference temperature for temperature dependent internal heat generation loading.

```
RIHG  n  [l]  [TITLE  title]
TABLE  ilod
V  T
```

n	The number of values of data in the table (must currently =1)
l	The starting location of the first input value in the table (default **1**=1).
ilod	The table reference number.
V	Rate of internal heat generation.
T	Reference temperature.

Notes

1. Load curves can be used to maintain or increment the RIHG as a nonlinear analysis progresses.
2. Automatic load incrementation under the NONLINEAR CONTROL data chapter cannot be used with RIHG loading.

User defined rate of internal heat generation (RIHG USER)

The USER defined rate of internal heat generation facility allows a user-supplied subroutine to be used from within LUSAS. This facility provides access to the LUSAS property data input via the RIHG USER data section and provides controlled access to the pre- and post-solution element processing via the user-supplied subroutine.

By default the routine is supplied as an empty routine with a defined interface that is unchangeable. The externally developed code should be placed into this routine which is then linked into the LUSAS system. Source code access is available to the interface routine and object library access is available to the remainder of the LUSAS code to enable this facility to be utilised. See Solver User Interface Routines.

Since user specification of rate of internal heat generation involves the external development of source FORTRAN code, as well as access to LUSAS code, this facility is aimed at the advanced LUSAS user.

The data section RIHG USER provides the interface to User supplied subroutines for computing the rate of internal heat generation.

```
RIHG USER n [TITLE  title]
TABLE  ilod
< U_i >_i=1,(n-1)  T
```

n The number of values of data in the table
ilod The table reference number.
U_i The user-defined input parameters
T Reference temperature.

Notes

1. NONLINEAR and TRANSIENT CONTROL chapters must be defined when using RIHG USER loading.
2. Load curves can be used to maintain or increment the RIHG USER as a nonlinear transient analysis progresses.
3. The number of input parameters must be equal to that specified on the data section header line (i.e. n). Failure to match the requested and supplied number of parameters will invoke a LUSAS error message.
4. Option 179 can be set for argument verification within the user routines.

Temperature Load Assignments

The TEMPERATURE LOAD ASSIGNMENTS data section is used to assign temperature dependent field loading (that is TDET and RIHG) to the associated elements.

```
TEMPERATURE LOAD ASSIGNMENTS  [TITLE  title]
```

Notes

1. Each load assignment list must start with this header.
2. The TEMPERATURE LOAD ASSIGNMENTS data section must be immediately followed by a TDET or RIHG assignment data section.
3. Load assignments must be used if temperature dependent field loads are specified.

TDET Load Assignments

The TDET assignment data section is used to assign the defined temperature dependent environmental temperature field loading to the associated elements.

```
TDET
L  LF  N  ilod
```

L The element number.

LF The element face number.
N The element (face) node number as defined in ELEMENT TOPOLOGY.
ilod The load table reference number (see Temperature dependent environmental temperature loading (TDET).

RIHG Load Assignments

The RIHG assignments data section is used to assign the defined temperature dependent and User defined rate of internal heat generation to the associated elements.

```
RIHG [USER]
L  N  ilod
```

L The element number.
N The element node number as defined in ELEMENT TOPOLOGY.
ilod The load table reference number (see Temperature dependent rate of internal heat generation (RIHG)).

Curve Definition

General curves may be defined in order to describe loads that vary with time in a dynamic analysis or to describe loads that vary with the angle around the circumference for a Fourier analysis. Curves may also be used to define the variation of load with increment number in a static analysis. A selection of pre-defined LUSAS system curves is available, or more generally, curves may be described completely by the user. The description of the curve (or curves) is done by using the CURVE DEFINITION command. The CURVE ASSIGNMENT command then associates load cases to a particular curve.

Curve Definition

The definition of a curve is controlled by the CURVE DEFINITION command.

```
CURVE DEFINITION  [lcurve] [USER | SINE | COSINE |
SQUARE]
```

for **USER** curve definition:

$$< t_i \quad F(t_i) >_{i=1,n}$$

for **SINE**, **COSINE**, **SQUARE** curve definition:

amplitude frequency [phase_angle t_0]

lcurve Load curve number (default is the order of CURVE DEFINITION input).
ti Interpolation variable (e.g. time, angle, increment number).
F(ti) The value of the function at **ti** (positive or negative as required).
amplitude The peak value of the wave (positive or negative as required).
frequency The frequency of the wave (i.e. the inverse of the time period).

phase_angle The offset in degrees applied to **t**.

t0 The value of the interpolation parameter at which the curve is activated.

Notes

1. The USER curve is assumed to be linear between defined points F(ti). It should be noted that each data point is input on a separate line. The USER curve is the default input.
2. Data should be input in order of increasing value of the interpolation parameter. The curve should, in general, only assume 1 value at each interpolation point. However, for Fourier static analysis a vertical segment may be used to define a step-function in the load distribution (see figure below).
3. If a point lies outside the USER defined curve no load is applied.
4. The LUSAS system curves are defined by:

   A curve amplitude

   F curve frequency

   ϕ phase offset (degrees)

   t_0 starting value.

   ❑ SINE curve

   $t < 0 \qquad F(t) = 0$

   $t > 0 \qquad F(t) = A\sin\left(f\left(t - t_0\right) + \phi\right)$

   For Fourier analysis: $F(t) = A\sin ft$

   ❑ COSINE curve

   $t < 0 \qquad F(t) = 0$

   $t > 0 \qquad F(t) = A\cos\left(f\left(t - t_0\right) + \phi\right)$

   For Fourier analysis: $F(t) = A\cos ft$

   Note. If f=0 then $F(t) = A\cos(\phi)$ for all t

   ❑ SQUARE curve

   $t < 0 \qquad F(t) = 0$

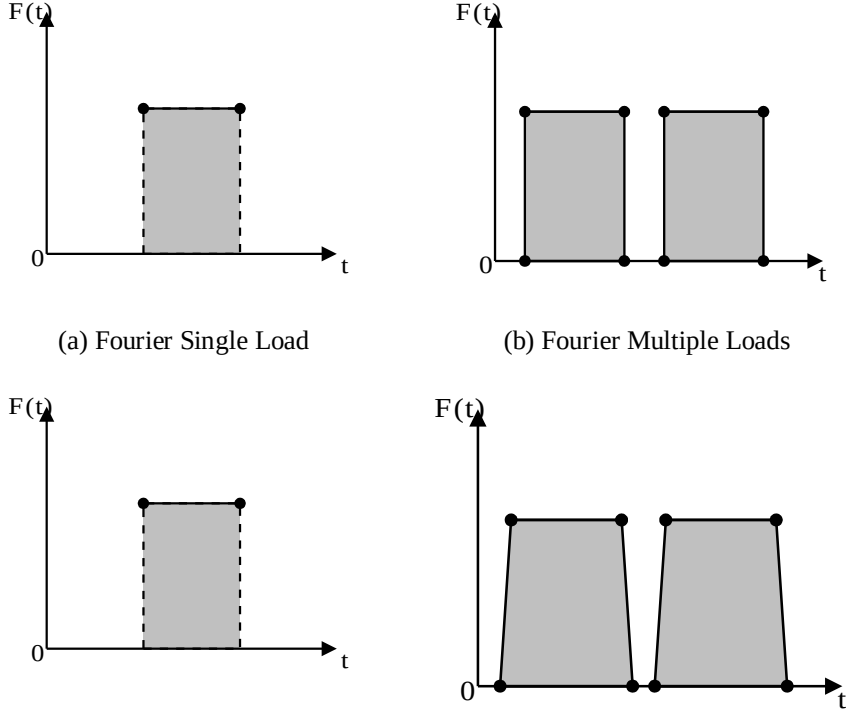
   $t > 0 \qquad F(t) = A \qquad \text{if} \qquad \sin\left(f\left(t - t_0\right)\right) > 0$

   $F(t) = -A \qquad \text{if} \qquad \sin\left(f\left(t - t_0\right)\right) < 0$

   $F(t) = 0 \qquad \text{if} \qquad \sin\left(f\left(t - t_0\right)\right) = 0$

   Note. If f=0 then $F(t) = A$ for all t.

5. Any number of curves may be specified. In the special case of an ENVT load in a thermal analysis, the temperature and heat coefficients are not applied.
6. For nonlinear static analysis the step increment number will be used to interpret the data. The INCREMENTATION data section must not be specified.
7. For Fourier analysis the load must only be applied over the angular range of 0° to 360°. Sine and cosine harmonics (without phase angle and constant specification) can be used to apply sinusoidal loading.
8. If load curves are used with NONLINEAR CONTROL the TERMINATION line must be specified to define the number of load steps to be applied. (See example below.)



(a) Fourier Single Load

(b) Fourier Multiple Loads

(c) Non-Fourier Single Load

(d) Non-Fourier Multiple Loads

Definition of Step Functions for a Fourier Static Analysis
(Black circles indicate points that must be defined)

Curve Assignment

The CURVE ASSIGNMENT command assigns loads defined by the LOAD CASE commands to a particular curve which has been described using the CURVE DEFINITION command.

```
CURVE ASSIGNMENT
L  fac  lcurve
```

L	Load case number (see *Notes*)
fac	The load factor to be applied to the load case
lcurve	The load curve number

Notes

1. If the CURVE ASSIGNMENT command is not utilised the load cases are applied as for a standard analysis.
2. If the CURVE ASSIGNMENT command is used, unassigned load cases will not be applied.
3. Curve data, both user and system, once input, cannot be overwritten, extended or amended. However, new load curves may be added. Once the CURVE ASSIGNMENT command has been used, load curves are activated and loading is applied via factored load cases. If the CURVE ASSIGNMENT command is respecified, all existing assignments are overwritten.
4. Each curve assignment must start on a new line.
5. The load case number is defined by the order in which the LOAD CASES are input. The first load case is defined as 1, the second 2 and so on. If, during the analysis, further load cases are defined then the internal load case counter is reset to 1. In this instance, existing load cases will be overwritten.
6. The standard function of prescribed displacements and spring stiffnesses input using the SUPPORT facility is preserved. If PDSP loading is used, this data overwrites previously input data. Note that PDSP input determines the incremental displacements and not the total displacements even if LOAD CURVES are utilised. To define total displacements using load curves TPDSP should be used.
7. General curves can be used in the following applications:

	Analysis Type	Interpolation Variable
dynamic	linear/nonlinear/explicit/implicit	current response time
transient	linear/nonlinear/explicit/implicit	current response time
nonlinear	static (not arc length)	increment number
Fourier	static	circumferential angle (degrees)

Example 1. Load Curves

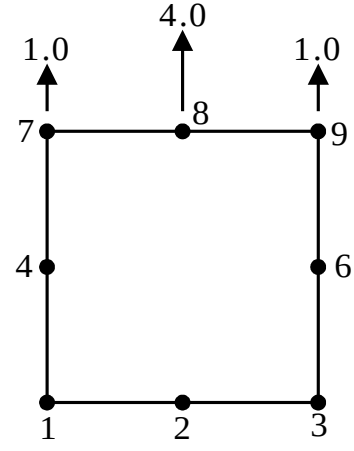
The commands used to double the intensity of a face load over a period of 10 units are

```
LOAD CASE
    FLD
    1  1  0     8.
CURVE DEFINITION   3   USER
    0   1 :assign a value of 1 at time/increment zero
    10  2 :assign a value of 2 at time/increment ten
CURVE ASSIGNMENT
    1  2  3     :assign load curve 3 with a factor of 2
to load case 1.
```

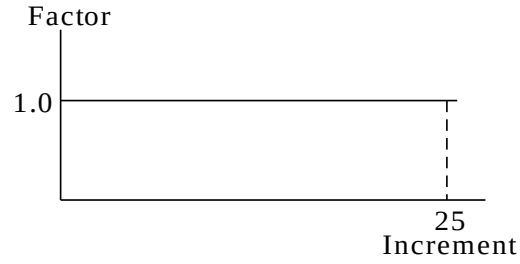
Example 2. Load Curves

The following example shows how to superimpose three load curves to give a combined loading of one 'dead' load and two 'live' loads in a static nonlinear analysis.

Unfactored loading:

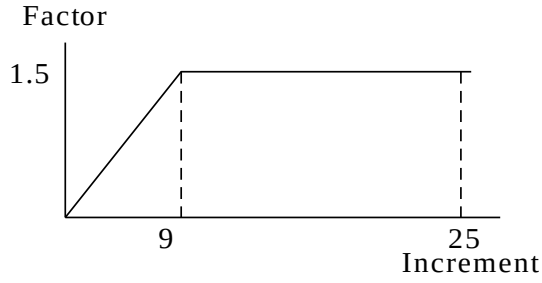


Load curves to be superimposed:



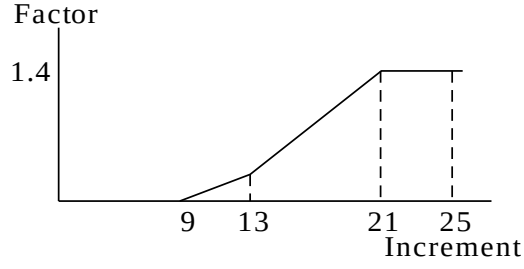
Load Curve 1

Dead load



Load Curve 2

Live load 1



Load Curve 3

Live load 2

Loading section from data file:

```
LOAD CASE
 CL
 7  9  2   0.0  1.0
 8  8  0   0.0  4.0
CURVE DEFINITION  1  USER
 1 1.0
```

```
 25  1.0
CURVE DEFINITION 2 USER
 1  0.0
 9   1.5
 25  1.5
CURVE DEFINITION 3 USER
 9   0.0
 13   0.4
 21   1.4
 25   1.4
CURVE ASSIGNMENT
 1  1.0  1
 1  1.0  2
 1  1.0  3
NONLINEAR CONTROL
 ITERATIONS  12
 CONVERGENCE  0.0  0.0  0.8  0.8
 OUTPUT  0  1  4
 TERMINATION  0  25
END
```

The use of Load Curves with Fourier Elements

Fourier loading should be input using the load curve commands to define its circumferential variation. The curve data must use q values lying between 0° and 360°. Both system and USER curve data may be utilised. For dynamic and harmonic response analyses the Fourier load component must be input explicitly.

Retained Freedoms

The RETAINED FREEDOMS data section is used to manually define **master** and **slave** freedoms. The definition of master and slave freedoms is required for Guyan reduced eigenvalue extraction. RETAINED FREEDOMS are also required to define the master freedoms for a superelement during the generation phase of an analysis involving superelements.

Within the Guyan reduction phase the number of master freedoms may automatically be generated to compute the correct number of iteration vectors required, **`nivc`**. This

depends upon the number of master freedoms already specified in the RETAINED FREEDOMS data section, **nmastr**. The number of automatically generated master freedoms would therefore be **namast** = **nivc** - **nmastr**.

RETAINED FREEDOMS [TITLE title]

N N_{last} N_{diff} < $type_i$ >$_{i=1,n}$

N N_{last} N_{diff} The first, last and difference between nodes of the series of nodes with identical retained freedoms.

typei The freedom type for each global freedom at a node.
=M for a master degree of freedom (default)
=S for a slave degree of freedom.

n The number of freedoms at a node for the structure being solved.

Notes

1. If a node number is repeated the new set of freedom types overwrites the previous set and an advisory message is printed out.
2. The master and slave codes must be separated by a space (for example M S M not MSM).
3. Master freedoms may also be selected automatically for Guyan reduction using the GUYAN CONTROL data chapter.
4. If the FREEDOM TEMPLATE data chapter has been utilised the freedom types specified in this section will relate to the modified freedom list.
5. If Option 131 is specified in your data file when retained freedoms are present, (Guyan reduction and superelement generation phases), the option is switched **off** and a warning message is issued. (Option 131 is switched on by default).

Output Control

The location of LUSAS output may be controlled in two ways:

- **Element output**
- **Node output**

Stresses and strains at Gauss points and nodes are output using the ELEMENT OUTPUT command. Nodal displacements and reactions are output using the NODE OUTPUT command.

The frequency of output is controlled using the OUTPUT data sections in the NONLINEAR, DYNAMIC, TRANSIENT and VISCOUS CONTROL data chapters. Further control is provided using the OPTION facility. The relevant output control options are:

No.	Output options
26	Reduce number of lines output in coordinates

No.	Output options
32	Suppress stress output but not stress resultant output
33	Output direction cosines of local Cartesian systems for interface
34	Output element stress resultants
39	Stress smoothing for rubber material models
40	Output nodal displacement increments for nonlinear analyses
42	Output nodal residual forces for nonlinear
44	Suppress expanded input data printout except load cases
46	Suppress page skip between output stages
55	Output strains as well as stresses
59	Output local direction cosines for shells
77	Output principal stresses for solids
115	Output displacements and reactions in transformed axes
116	Suppress the output of internal constraint forces
143	Output shear forces in plate bending elements. This option will produce results on an element-by-element basis, but will not include the shear stresses in the averaged nodal results.
144	Output element results for each load case separately
147	Omit output phase
169	Suppress extrapolation of stress to nodes for semiloof shell
181	Output required results in Polar system for harmonic response analysis
208	Write plot file for pre version 10.0 LUSAS Modeller
259	Save nodal stresses in plot file to avoid stress extrapolation in LUSAS Modeller

Element Output Control

The ELEMENT OUTPUT data section is used to define the order and contents of the element output.

```
ELEMENT OUTPUT  [ASCENDING | SPECIFIED | SOLUTION]
[TITLE  title]

L Llast Ldiff  iout  [iavgp]  [ifrmgp] [inthis]
```

L Llast Ldiff The first element, last element and difference between elements of the series of elements with identical output control.

iout The element output number
=0 for no output
=1 for nodal output

=2 for Gauss point output
=3 for nodal and Gauss point output

iavgp The averaged nodal value group number (see *Notes*).

ifrmgp Parameter to control the grouping of the summation of element forces and moments for the model summary.

inthis Parameter to control element output to the history file. Options are:
= 0 for no output.
= 1 for output.

Notes

1. If this data chapter is excluded nodal results will be output for all elements in ascending order.
2. Element numbers not specified are not output.
3. For linear problems ELEMENT OUTPUT control must be placed immediately after all loading data. The output control cannot be varied from load case to load case.
4. For nonlinear/step-by-step problems ELEMENT OUTPUT control may be placed after each LOAD CASE to change output between load increments. Once specified the ELEMENT OUTPUT control will remain active throughout the problem until respecified.
5. For load combinations ELEMENT OUTPUT control may be respecified after any LOAD COMBINATIONS data chapter. ELEMENT OUTPUT then controls the output of the preceding load combinations and remains in force unless respecified further on.
6. If ELEMENT OUPUT is not specified after LOAD COMBINATIONS, the ELEMENT OUTPUT control currently in force for LOAD CASES is used.
7. For enveloping ELEMENT OUTPUT control may be specified after the ENVELOPE FINISH data line to control the output of results from an envelope.
8. Elements may be output in ascending, specified or solution order. The specified order is the order in which elements appear or are generated in the ELEMENT OUTPUT PECIFIED data chapter. It should be noted that when assembling the output order LUSAS, firstly, groups together all elements with the same averaged nodal value grouping, secondly, groups all similar elements (that is, all solids shells, etc.) and thirdly, sorts out the specified order.
9. If averaged nodal values are required then a positive integer must be specified for **iavgp**. This allows complete control to be exercised over which element nodal values are averaged together; a particularly useful facility where stresses are in local directions, i.e. shells. Any number of different groups may be specified and only elements with the same group number are averaged together.
10. The parameter **ifrmgp** should be specified as a positive integer that defines the group number for the summation of element forces and moments in the model summary.

11. The averaged nodal values produced with ELEMENT OUTPUT do not include the thin isoflex plate shear stresses that appear in the element-by-element output with Option 143.

12. All element data is always written to the LUSAS Modeller post processing file.

Node Output Control

The data section NODE OUTPUT is used to control the location and content of the nodal output.

```
NODE OUTPUT  [TITLE  title]

N Nlast Ndiff  int [inthis]
```

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical output control.

int The node output number
=0 for no output
=1 for displacements at nodes
=2 for reactions at nodes
=3 for displacements and reactions at nodes.

inthis Parameter to control element output to the history file. Options are:
=0 for no output.
=1 for output.

Notes

1. If this data chapter is excluded nodal results will be output for all nodes in ascending order.

2. Node numbers not specified are not output

3. For linear problems NODE OUTPUT must be placed immediately after all loading data. The output control cannot be varied from load case to load case.

4. For nonlinear/step-by-step problems, NODE OUTPUT control may be placed after each LOAD CASE to change output between load increments. Once specified, the NODE OUTPUT control will remain active throughout the problem until respecified.

5. For load combinations, NODE OUTPUT control may be respecified after any LOAD COMBINATION data chapter. NODE OUTPUT then controls the output of the preceding load combinations and remains in force unless respecified further on.

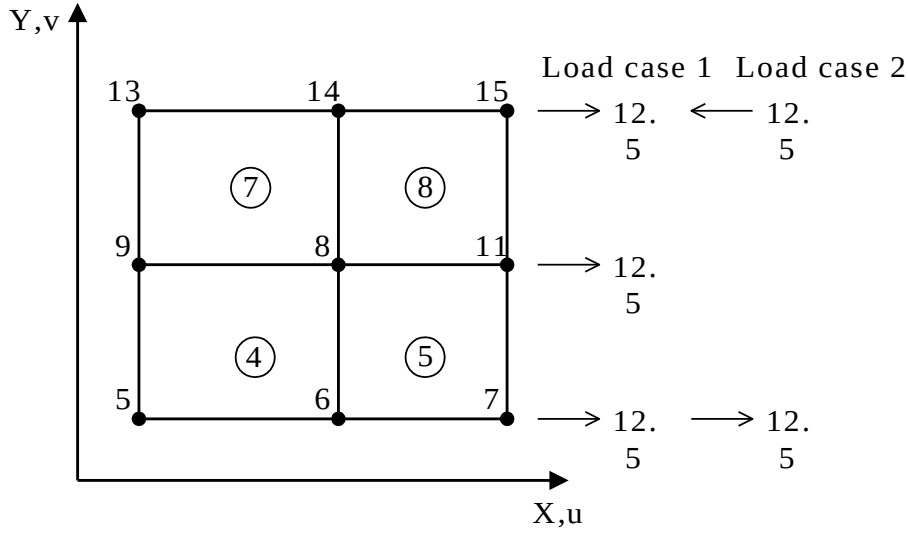
6. If NODE OUTPUT is not specified after LOAD COMBINATIONS, the NODE OUTPUT control currently in force for LOAD CASES is used.

7. For enveloping, NODE OUTPUT control may be specified after the ENVELOPE FINISH data line to control the output of results from an envelope.

8. Note that for field analyses, the displacement output (`int`=1 or 3) will be replaced by field values, thermal gaps or radiation surface flows.

9. All nodal data is always written to the LUSAS Modeller post processing file.

Example 1. Element and Node Output with Load Combination

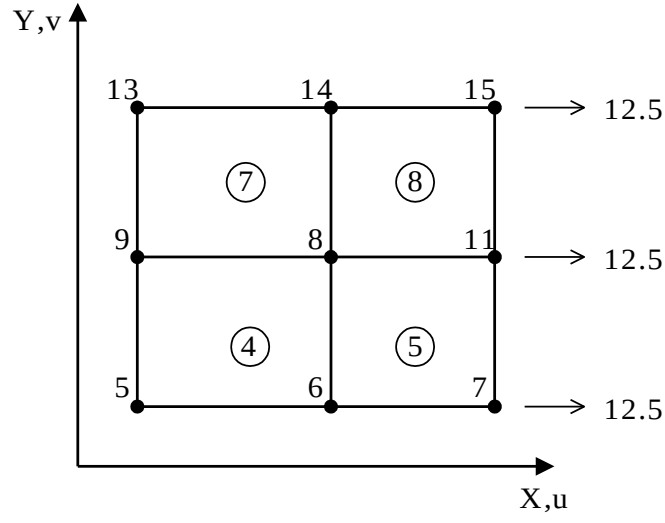


Plane stress problem:

```
LOAD CASE
 CL
  7 15 4  12.5
LOAD CASE
 CL
  7 0 0    12.5
  15 0 0 -12.5
ELEMENT OUTPUT ASCENDING
 4 5 1 1
 8 0 0 1
NODE OUTPUT
 5 7 1      3
 13 15 1  3
LOAD COMBINATION
```

```
 5.0 1
 2.5 2
LOAD COMBINATION
 3.0 1
 4.5 2
ELEMENT OUTPUT SPECIFIED
 4 7 3 3
 5 8 3 2
LOAD COMBINATION
 1.5 2
-1.5 1
ELEMENT OUTPUT ASCENDING
 5 0 0 1
```

Example 2. Element and Node Output for a Single Load Case (Linear)




```
LOAD CASE
 CL
  7 15 4 12.5
ELEMENT OUTPUT
```

```
C Gauss point stresses for element 4
4 0 0 2
NODE OUTPUT
C Displacement and reaction output
C for nodes 5, 6 and 7
 5 7 1      3
```

Example 3. Element and Node Output for Single Load Case (Nonlinear)

```
LOAD CASE
 CL
  7 15 4  12.5
ELEMENT OUTPUT
 4 0 0 3  : gives Gauss point and nodal stresses for
element number 4
          : the omition of the NODE OUTPUT command gives
displacements and
          : reactions for all nodes
NONLINEAR CONTROL
```

Load Combinations

The data section LOAD COMBINATIONS may be used to combine the results obtained from individually defined load cases.

```
LOAD COMBINATIONS  [N]  [TITLE  title]
fac  L
```

N The load combination number (default sequential, starting with load combinations 1).

fac The multiplication factor for load case.

L The load case to be factored.

Notes

1. Only results from LOAD CASES initially specified in the data input may be factored.
2. There is no limit on the number of LOAD COMBINATIONS which may be processed.

3. There is no limit on the number of LOAD CASES which may be factored to create a LOAD COMBINATION.
4. If a load case is specified more than once in the LOAD COMBINATION data the factors are accumulated.
5. The load case number is implicitly assigned in the order of declaration of the LOAD CASES in the datafile. It must not be specified with the LOAD CASE data section.
6. A PLOT FILE cannot be specified after LOAD COMBINATION.
7. The results of a LOAD COMBINATION and the combination data itself is not transferred to the LUSAS Modeller post-processing file.
8. LOAD COMBINATION can also be carried out in LUSAS Modeller.

Enveloping Results

The ENVELOPE data section may be used to extract the maximum and minimum values from a series of enveloped linear analyses.

A particular load envelope is specified between the commands ENVELOPE START and ENVELOPE FINISH. The LOAD COMBINATION commands appear between the envelope commands. Output control for the envelope is specified immediately after the ENVELOPE FINISH command. Specification of nodal and element output within the envelope will be applied to the preceding LOAD COMBINATIONS.

```
ENVELOPE  {START | FINISH}
```

Notes

1. Enveloping is available for all elements in the LUSAS Element Reference Manual.
2. The enveloping facility may be used in conjunction with the RESTART facility if required.
3. Enveloping must be used in conjunction with the LOAD COMBINATION facility.
4. Enveloping can also be carried out in LUSAS Modeller.

Superelements

Large finite element models may be divided into smaller, more manageable components which are added together to form the complete structure in a subsequent analysis. These components are referred to as ‘superelements’. A superelement may be defined as an assembly of individual elements together with a list of master freedoms that will represent the superelement once it has been reduced. The SUPERELEMENT family of commands provides this facility in LUSAS for linear static, eigenvalue, nonlinear and transient dynamic analyses. An option also exists to evaluate the natural frequencies of a large structure from the eigen solutions of the individual superelements using modal synthesis.

In addition a component for which the user modal data (frequencies and eigenvectors) is available can be utilised to obtain the natural frequencies of a large structure. This component will be referred to as a user modal superelement. A detailed explanation of these facilities can be found in the *LUSAS Theory Manual*.

Analyses involving superelements may be divided into three basic stages:

- Opening databases and creating superelement data.
- Using superelements and/or user modal data in an analysis (i.e. solving the complete structure). This stage could involve user modal data only if superelements are not used.
- Recovering displacements and stresses within the superelements.

The superelement data may be created in one or several analyses and stored in several databases. Once generated, the superelements may be used in either static, natural frequency, nonlinear or transient dynamic analysis. The recovery procedure permits the displacements, stresses and strains of a superelement to be recovered. Additional concentrated loads and prescribed displacements are permitted during the combination pass of an analysis. In the create phase, both static and natural frequency analyses of an individual superelement may be performed to validate the data.

The following restrictions apply to the use of superelements:

1. If nonlinear analysis are undertaken the super elements will behave linearly.
2. Superelements cannot be generated from Fourier elements.
3. User modal data may only be used for natural frequency analyses
4. Material properties must be constant. If temperature dependent properties are input then the values corresponding to the initial temperature will be used.
5. Superelement generation/recovery is limited to one superelement per analysis run.
6. When a superelement connects to other non-superelements, the connecting node locations associated with the non-superelements must have the same relative locations as originally used in generating the superelement. In addition, the Cartesian sets for the nodes must coincide.
7. If superelements are used without any standard elements the FREEDOM TEMPLATE data chapter must be included.
8. If superelements are used that have more freedoms than the standard elements, and any values are to be prescribed for these freedoms, then the FREEDOM TEMPLATE data chapter must be included. These values may be defined in SUPPORT CONDITIONS, CONSTRAINT EQUATIONS, RETAINED FREEDOMS or LOADCASE data chapters.
9. User modal data eigenvectors must be in the structure global axes.

How to use Superelements

The general approach for using the SUPERELEMENT facility in LUSAS is explained through a description of input data files required at each stage of a typical analysis

using superelements. The output files provided at each stage are also listed and their use in subsequent stages of the analysis is defined. The figure on Recovery Phase gives a schematic diagram of a complete superelement analysis together with the input and output files at each stage. An example of data files required for each phase of the analysis is provided on Examples of Using Superelements.

Creating Superelements

A typical data file for creating superelements would contain SUPERELEMENT ASSIGN and SUPERELEMENT DEFINE commands. The master freedoms for the superelement would be specified using the RETAINED FREEDOMS data chapter. This analysis would then create:

- A superelement database
- An output file
- A restart file (optional)

The contents of the files created would depend upon the control parameters specified under the sub-commands of SUPERELEMENT DEFINE. Only one superelement can be generated in a single analysis.

Using Superelements in an Analysis

A typical data file for using superelements in an analysis to solve the complete structure would contain the SUPERELEMENT USE command and may also contain SUPERELEMENT DEFAULT or DISTRIBUTE commands. These commands would reference databases created in the superelement creation phase. Additional elements, boundary conditions or loading may be specified at this stage. This analysis would then create:

- Superelement result database(s)
- An output file
- A restart file (optional)
- A LUSAS Modeller plot file (optional)

Several superelement databases may be used in this stage of the analysis. In addition user modal superelements may be included to form the complete structure using the SUPERELEMENT MODAL_DATA command. User modal superelements can also be used in isolation (see Utilising User Modal Data in an Analysis below). If a LUSAS Modeller plot file is created it will NOT contain the superelement results.

Recovering Superelement Results

A data file for recovering superelement results using output from the previous analysis stage would contain the SUPERELEMENT RECOVER command and may also contain a SUPERELEMENT DEFAULT command. These commands would reference

output databases created during solution of the complete structure. This analysis would then create:

- An output file
- A restart file (optional)
- A LUSAS Modeller plot file (optional)

If the superelement generation restart file is not available recovery may be performed using the original data file. Only one superelement may be recovered in each analysis.

Utilising User Modal Data in an Analysis

A typical data file for utilising user modal data in an analysis to solve the complete structure would contain the SUPERELEMENT MODAL_DATA command. This command would reference a neutral file containing the component connectivity nodes and freedoms, frequencies and references to result files containing the eigenmodes. Additional user modal data components, elements, superelements or boundary conditions may be specified at this stage. The analysis would then create:

- An output file
- A restart file (optional)
- A LUSAS Modeller plot file (optional)

If a LUSAS Modeller plot file is created it will NOT contain the user modal data results.

A Typical Superelement Analysis

A typical analysis sequence to generate two superelements and use them with a user modal superelement and a group of LUSAS elements in the complete structure would be as follows (see figure on Recovery Phase):

Generation Phase

Analysis 1

Input	Output
define1.dat	define1.out
	define1.mys (optional)
	define1.rst (optional)
	define1.sfr (rename to recov1.sfr or delete)
	dbname1.sda

Analysis 2

Input	Output
define2.dat	define2.out

Analysis 2	Input	Output
		define2.mys (optional)
		define2.rst (optional)
		define2.sfr (rename to recov2.sfr or delete)
		dbname2.sda

Solution Phase

Analysis 3	Input	Output
	use1.dat	use1.out
	dbname1.sda	use1.rst (optional)
	dbname2.sda	use1.mys (optional)
	use1.def	dbname1.srs
	use1.adp	dbname2.srs

Recovery Phase

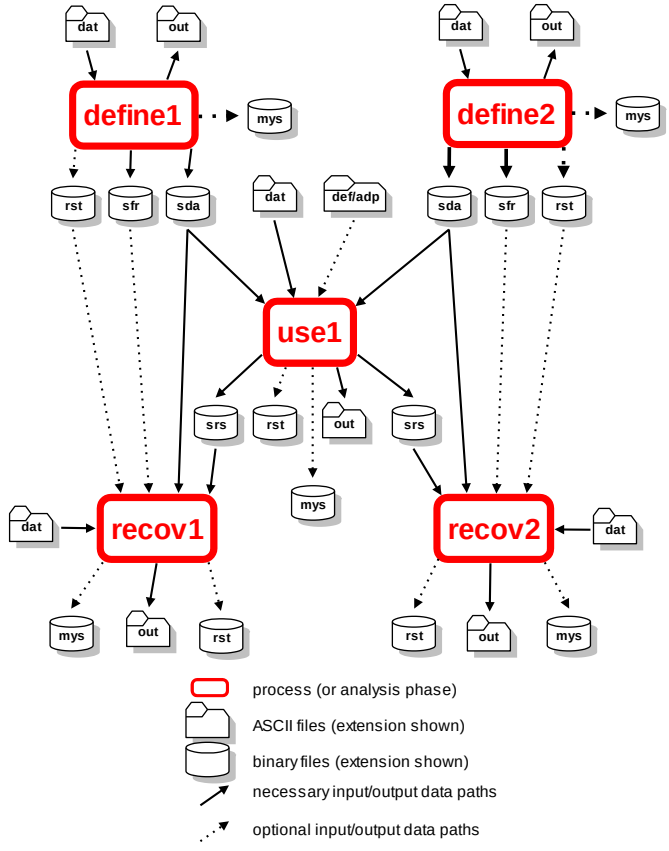
Analysis 4	Input	Output
	recov1.dat	recov1.out
	define1.rst (optional)	recov1.mys (optional)
	recov1.sfr (optional)	recov1.rst (optional)
	dbname1.sda	dbname1.srs

Analysis 5	Input	Output
	recov2.dat	recov2.out
	define2.rst (optional)	recov2.mys (optional)
	recov2.sfr (optional)	recov2.rst (optional)
		dbname2.sda
		dbname2.srs

The file extensions .sda and .srs relate to superelement data and results databases respectively. The extension .sfr indicates a file containing the reduced frontal matrices. If the appropriate .sfr file is accessible during the recovery phase, the solution is recovered from the back substitution of the equations leading to a more efficient solution. To take advantage of this facility the .sfr file must have the same prefix as the

data file used in the recovery, i.e. copy define1.sfr to recov1.sfr in above example. If a superelement restart file (.rst) is not available, recovery may be performed by redefining the superelement in the datafile (i.e. as in the superelement creation phase).

Note. The .mys, .rst, .sfr, .sda and .srs files are all binary files. The .dat and the .out files are ASCII files which can be read and edited directly by the user. The user modal data files (.def and .adp) are ASCII files.



Main Stages of an Analysis Involving Superelements

Component Mode Synthesis Using Superelements

Component mode synthesis provides a method of evaluating the natural frequencies of a large structural system by utilising the eigen solutions of the component parts. Additional generalised freedoms (or modal coordinates) are introduced which enable an enhanced representation of the reduced mass and stiffness to be computed for each superelement. The superelement matrices are then combined and the eigen-solution for

the complete structure is evaluated. More details can be found in the *LUSAS Theory Manual*.

The eigenvalues evaluated using this method of analysis will always be an upper bound on the corresponding exact values of the system. The accuracy of each eigenvalue and corresponding eigenvector is dependent upon how accurately the boundary conditions are represented in the eigen-analysis of each of the components.

The component mode reduction can be carried out whilst defining a superelement, or as an updating procedure if the superelement has already been defined. During this phase LUSAS calculates the natural frequencies of the superelement. Using this eigen-solution the reduced stiffness and mass are computed for the superelement, in terms of the master freedoms and generalised coordinates. For these computations the required number of generalised coordinates must be defined in the data input.

- **Generalised coordinates** are extra (fictitious) degrees-of-freedom that are utilised to enhance the stiffness and mass of the superelement and allow a more accurate dynamic (eigen or transient) solution of the complete structure to be computed without the specification of extra retained (or master) freedoms.

In addition, the modal supports for the master freedoms must also be specified as 'fixed', 'free' or 'spring stiffness and mass'.

For many structures the alternatives 'fixed' or 'free' are not sufficient for accurate modelling. In order to improve modelling it is possible to:

- Introduce spring stiffnesses and masses on the exterior freedoms to represent the effects of the remaining structure. This requires you to estimate the stiffness and mass of the remaining structure.
- Automatically reduce the stiffness and mass of the remaining structure to the master freedoms. This can be achieved in two ways:
  - once the component mode reduction has been carried out for each superelement the complete structure can be assembled and the natural frequencies computed. At this stage it is also possible to automatically distribute the mass and stiffness of the complete structure to each superelement.
  - alternatively the stiffness and mass can be distributed before any natural frequencies are calculated at the superelement level. Once this redistribution has been carried out the natural frequencies of the individual superelements can be computed more accurately.

Opening Superelement Files

Superelement databases are opened using the SUPERELEMENT ASSIGN command.

```
SUPERELEMENT {ASSIGN dbname filename | DEFAULT
dbname}
```

ASSIGN Opens a new superelement database.
DEFAULT Changes the default superelement database.
dbname Internal name of database.
filename External name of file containing database.

Notes

1. If a filename is not provided, the filename will default to dbname.
2. Several superelement databases may be used at any one time. Use of the DEFAULT option changes the default database for subsequent operations.
3. ASSIGN sets the database defined as the default database.

Superelement Creation

The superelement to be created will contain information on element topology and usually requires specification of boundary freedoms, loading and reduction options. The SUPERELEMENT DEFINE command defines the name and title associated with the superelement.

Superelement Definition

SUPERELEMENT DEFINE defines the superelement name and internal database that is used to store the superelement information.

```
SUPERELEMENT DEFINE  sname  [dbname]  [TITLE  title]
```

sname Superelement name.
dbname Internal name of database.

Notes

1. Each superelement in an analysis must have a unique identification name.
2. If **dbname** is not specified then either:
3. The current default database will be used if available, or
4. The database will default to **sname** if a default database is not available.

Constants

The CONSTANTS data section defines the control parameters for the superelement creation phase. A list of valid names defines the output required.

```
CONSTANTS  [LOAD]  [STIF]  [MASS]  [STAT]  [EIGN]
```

LOAD Save the reduced load vector.
STIF Save the reduced stiffness matrix.
MASS Save the reduced mass matrix.

STAT Perform a static analysis using superelement.
EIGN Perform an eigen-analysis using superelement.

Notes

1. If the CONSTANTS command is not specified LOAD and STIF will be invoked by default. However, if Option 90 is set then LOAD, STIF and MASS will be the default control parameters.
2. If the command GENERALISED COORDINATES is specified the following default operations will be carried out:

    - The reduced mass matrix will be saved.
    - An eigen-analysis will be performed to compute the eigenvectors required.
    - The enhanced reduced stiffness and mass will be computed and saved.

Output

The OUTPUT command defines the information to be presented in the results output file.

```
OUTPUT  [MAST]  [LOAD]  [STIF]  [MASS]  [DRCO]  [CORD]
```

MAST Master freedom description.
LOAD Reduced load vector.
STIF Reduced stiffness matrix.
MASS Reduced mass matrix.
DRCO Nodal direction cosine matrices.
CORD Nodal coordinates.

Notes

1. By default, only a summary of the superelement data will be provided.

Generalised Coordinates

The GENERALISED COORDINATES command defines the number of generalised coordinates to be utilised in the exterior freedom set for the superelement. These additional freedoms allow an enhanced representation of the superelement mass and stiffness to be computed thereby providing a more accurate solution when the complete structure is analysed.

```
GENERALISED COORDINATES  m
```

m Number of generalised coordinates

Notes

1. If the GENERALISED COORDINATES command is specified an eigenvalue analysis will be performed to compute `max(m,nroot)` eigenvalues and corresponding eigenvectors using either Subspace iteration or Guyan reduction.
2. If the eigenvalue solution method is not specified subspace iteration will be used by default.

Modal Supports

The MODAL SUPPORT command defines the support conditions for the master freedoms for local superelement eigen-analysis. This information is also used for computation of the eigen-modes associated with generalised coordinates. Further additional master freedoms may be specified for a local eigenvalue analysis using Guyan reduction.

MODAL SUPPORT [STIF] [MASS]

N N_{last} N_{diff} < $type_i$ >$_{i=1,n}$ < $stif_i$ >$_{i=1,n}$ < $mass_i$ >$_{i=1,n}$

STIF Stiffness flag.

MASS Mass flag.

N N_{last} N_{diff} The first node, last node and difference between nodes of the series of nodes with identical modal supports.

typei Modal support code for each freedom at the nodes. Valid types are:
R - Master freedom restrained.
F - Master freedom free.
S - Spring support for master freedom.
G - Denotes interior freedoms as masters for local eigenvalue analysis (see *Notes*).

stifi Stiffness for spring support.

massi Mass for spring support.

n Total number of degrees-of-freedom at the node.

Notes

1. By default, master freedoms are restrained ('R') in a local eigenvalue analysis.
2. MODAL SUPPORT has no influence in static analyses.
3. Retained freedoms of type 'G' should only be specified if Guyan reduction is being used to compute the eigen-solution of the superelement. The freedoms are not used in subsequent superelement operations.
4. The FREEDOM TEMPLATE data chapter must be used when MODAL SUPPORTS are specified.

Using Superelements

Superelements are utilised for two reasons:

- to solve the complete structure
- to redistribute the masses and stiffnesses for eigen-analyses.

The SUPERELEMENT USE command is used in both of these cases.

When solving the complete structure additional elements, loading, boundary conditions and user modal data may also be added directly with the complete structure. The complete structure is defined as the combination of the lowest level superelements. When the complete structure is solved, the displacements relating to all master freedoms are written to the appropriate superelement database(s).

The masses and stiffnesses of the complete structure can be redistributed to the boundaries of each of the superelements using the SUPERELEMENT DISTRIBUTE COMMAND (see Distribution of Mass and Stiffness in Modal Synthesis). In order to do this the complete structure must be formed by utilising the SUPERELEMENT USE command.

```
SUPERELEMENT USE  sname  dbname
```

sname Name of superelement to be solved.
dbname Database where superelement is stored.

Notes

1. Several superelements may be included in the complete structure.
2. If dbname is not specified then either:
3. the current default database will be used if available, or
4. the database will default to sname if a default database is not available.
5. If SUPERELEMENT ASSIGN has not been defined the filename which contains the superelement database, dbname, will be assumed to be the filename.
6. Additional loads and boundary conditions may only be added to the nodes specified as masters in the creation phase.
7. Additional load conditions permitted are concentrated loads (CL) only. Elements loads are not permitted at this stage.

Utilising User Modal Data

The SUPERELEMENT MODAL_DATA command is used to define modal data for a component of the complete structure. This command is only valid for natural frequency analyses.

```
SUPERELEMENT MODAL_DATA  uname
OUTPUT  {STIFF | MASS}
```

uname Name of neutral file containing user modal data.
STIF Reduced stiffness matrix to be output.
MASS Reduced mass matrix to be output.

Note

1. Several user modal superelements may be included in the complete structure.

User Modal Data Neutral File

The user modal data neutral file (with a file extension .def) has two lines for a title (packet type 25) followed by two lines for a subtitle (packet type 26 usually date, time and version). This is followed by the user modal data (packet type 27). This file can be in free format or in fixed format and the format type is specified on the first line of the neutral file. The results files (with a file extension of .adp) utilised in this neutral file must also be in this format.

The file format type for the neutral file and associated result files is specified on the first line of the neutral file as free or fixed. In free format you can input data in any column with a spacing between descriptors or numbers. fixed format requires the numbers to be input in certain columns. If this line is omitted or an unrecognised format type is encountered free format is assumed.

FORMAT Format (A)

```
   FORMAT
FORMAT
```

FORMAT = File format type (FREE or FIXED)

The header line for each data packet contains the following information:

HEADER Fixed Format (I2,8I8) or FREE format

```
   IT  ID      IV      KC      N1      N2      N3      N4      N5
```

IT = packet (or Entity) type
ID = identification number. A "0" value means not applicable
IV = additional ID. A "0" value means not applicable
KC = line count (number of lines of data after the header)
N1 to **N5** Supplemental integer values used and defined as needed.

PACKET TYPE 25: TITLE

HEADER Fixed Format (I2,8I8) or FREE format

```
   25  0      0      1
   ID  = 0   (n/a)
   IV  = 0  (n/a)
   KC  = 1
```

USER TITLE Format (A)

```
   TITLE
```

TITLE = identifying title, may be up to 80 characters

PACKET TYPE 26: SUMMARY DATA

HEADER Fixed Format (I2,8I8) or FREE format

```
   26  0      0      1
```

ID = 0 (n/a)
IV = 0 (n/a)
KC = 1

SUMMARY DATA Format (A)

```
   DATA
```

DATA = character string containing Date:Time:Version

PACKET TYPE 27: USER MODAL DATA

HEADER Fixed Format (I2,8I8) or FREE format

```
   27  0      0      KC     N1     N2     N3
```

ID = 0 (n/a)
IV = 0 (n/a)
N1 = maximum number of freedoms to a node
N2 = total number of boundary nodes i.e. connectivity to complete structure
N3 = number of eigenmodes
KC = 1 + N2 + (N3 + 4) / 5 + N3

TEMPLATE Fixed Format (20A4) or FREE format

FRETYP

FRETYP = freedom types at a node (N1 values specified)
For valid freedom types (see Freedom Template)

NODAL FREEDOM Fixed Format (10I8) or free format

NNODE NF1 NF2 NF3 NF4 NF5 NF6 NF7 NF8 NF9

NNODE = boundary node number - this node must exist in the structure and the results files

NF1 to **NF9** nodal freedoms (results column number in results file) - do not specify more than N1 freedoms

FREQUENCIES Fixed Format (5E16.9) or FREE format

FREQ

FREQ = natural frequencies for each mode

RESULTS FILE Format (A)

EIGFLE

EIGFLE = name of results file for this mode number
- this file contains the mass normalised eigenvectors and can be in free or fixed
format as specified on the first line in the neutral file, this line is repeated for each
mode

See Examples of Using Superelements for an example of a user modal data neutral file.

User Modal Data Results Files

The user modal data results file (with a file extension .adp) has one line for a title followed by a line for nodal information. This is followed by two lines for subtitles,

followed by nodal displacements (a line for each node). This file must be in free or fixed format as specified in the user modal data neutral file.

RESULTS FILE DATA Format (A)

TITLE

```
TITLE
```

TITLE = identifying title, may be up to 80 characters

NODAL INFORMATION Fixed Format (2I9,E15.6,2I9) or free format

```
NNODES    MAXNOD    DEFMAX    NDMAX NWIDTH
```

NNODES = number of nodes
MAXNOD = highest node number
DEFMAX = maximum absolute displacement
NDMAX = node number where maximum displacement occurs
NWIDTH = number of result columns after NODID for nodal freedom displacements

SUBTITLE1 Format (A)

```
SUBTITLE
```

SUBTITLE = identifying subtitle, may be up to 80 characters

SUBTITLE2 Format (A)

```
SUBTITLE
```

SUBTITLE = identifying subtitle, may be up to 80 characters

DISPLACEMENTS Fixed Format (I8,5E13.7) or free format (for each node)

```
NODID     (DATA(J),J=1,NWIDTH)
```

NODID = node number
DATA = nodal freedom displacements

See Examples of Using Superelements for an example of a user modal data results file.

Recovery of Superelement Data and Results

The SUPERELEMENT RECOVER command permits superelement results to be recovered from the database. This process involves reflation of the reduced solution for a superelement to obtain the solution of the constituent elements. The superelement database will contain the displacements relating to the exterior freedoms computed in the solution of the complete structure. In the recovery stage of the analysis these displacements will be used to compute internal displacements, stresses and strains for the superelement.

```
SUPERELEMENT RECOVER  sname  dbname
CONSTANTS  {STAT | EIGN | NONL inc | DYNA nt }
```

sname Name of superelement to be solved.
dbname Database where superelement is stored.
STAT Recover information from static analysis (see *Notes*).
EIGN Recover information from eigen-analysis (see *Notes*).
NONL inc Recover information from nonlinear analysis on specified increment.
DYNA nt Recover information from dynamic analysis on specified time step.

Notes

1. If **dbname** is not specified then either:

   The current default database will be used if available,

   or

   database will default to **sname** if a default database is not available.
2. If SUPERELEMENT ASSIGN has not been defined the filename which contains the superelement database, dbname, will be assumed to be the filename.
3. If the CONSTANTS command is not specified the results from the most recent analysis will be recovered.
4. A more efficient solution can be obtained if the appropriate file containing the reduced frontal matrices is utilised in the recovery stage. These files can be recognised by the extension .sfr and are produced during the superelement creation phase. To take advantage of this facility, the previously created .sfr file must be renamed to fname.sfr, where fname is the name of the data file used in the recovery. Using this approach the re-assembly of the reduced frontal matrices is avoided and the solution is achieved directly from the back substitution. However, it should be noted that the .sfr files are generally quite large and may require a considerable amount of storage.
5. When recovering problems in which generalised coordinates have been used the sfr file should be used in the recover process to maintain consistancy.

Distribution of Mass and Stiffness in Modal Synthesis

Generally, the master freedoms of a superelement are assumed to be either fixed or free. A better approximation is usually obtained by utilising point masses and stiffnesses to represent the effects of the remaining structure. This is accomplished by reducing the mass and stiffness of the complete structure to the boundary nodes of each chosen superelement. To achieve this, the master freedom codes for each superelement are modified and a Guyan reduction analysis is performed. This provides the reduced stiffness and mass from which the superelement stiffness and mass are subtracted to give the boundary mass and stiffness.

The SUPERELEMENT DISTRIBUTE command specifies that an analysis is to be performed to distribute mass and stiffness from the complete structure to each superelement in modal synthesis analysis. The complete structure must be formed by utilising the SUPERELEMENT USE command for each superelement that is to be included.

```
SUPERELEMENT DISTRIBUTE  sname  dbname
```

sname Name of superelement to be solved.
dbname Database where superelement is stored.

Notes

1. Several superelements may be included in the complete structure (see Using Superelements).
2. If **dbname** is not specified the data will be recovered from the default database set using the SUPERELEMENT ASSIGN/DEFAULT command.
3. If SUPERELEMENT ASSIGN has not been defined the filename which contains the superelement database, **dbname**, will be assumed to be the filename.

Examples of Using Superelements

This section provides an example of the data files required to execute the basic phases of an analysis using superelements.

Example 1. Superelement Generation Phase

```
PROBLEM TITLE Superelement generation phase
BEAM ELEMENT TOPOLOGY
 FIRST  1  1  2
 INC    1  3  3  3
NODE COORDINATES
 FIRST  1  0  0
```

```
 INC    1  5  0  2
 INC    3  0  5    5
BEAM GEOMETRIC PROPERTIES
 1   0.01  0.0001  1.0
GEOMETRIC ASSIGNMENTS
 1  3  1  1
MATERIAL PROPERTIES
 2  210E9  0.3
MATERIAL ASSIGNMENTS
 1  3  1  2
FREEDOM TEMPLATE
 U  V  THZ
SUPPORT NODES
 1  7  3  R  R  R
RETAINED FREEDOMS
 2  8  3  M  M  M
C
C  Open a superelement database SEDBSE in an external
file SEEXTF
SUPERELEMENT ASSIGN   SEDBSE      SEEXTF
C
C  Define the name of this superelement as SENAME and
store in the default database SEDBSE
SUPERELEMENT DEFINE   SENAME      TITLE First
superelement
C
C  Save the reduced stiffness and mass for SENAME in
SEDBSE
CONSTANTS       STIF  MASS
C
C  Present the master freedom description and reduced
stiffness in the results output file
OUTPUT                MAST  STIF
```

```
C  This restart file will be referred to as SEREST
RESTART WRITE
END
```

Example 2. Superelement Mass and Stiffness Distribution Phase

```
PROBLEM TITLE Superelement mass and stiffness
distribution phase
C
C  Open the superelement database SEDBSE in the external
file SEEXTF
SUPERELEMENT ASSIGN   SEDBSE      SEEXTF
C
C  Include the superelement SENAME (which is stored in
database SEDBSE) into the analysis.
SUPERELEMENT USE      SENAME      SEDBSE
C
C  Include other superelements into the complete
structure
C
C  Distribute mass and stiffness to superlement
SUPERELEMENT DISTRIBUTE     SENAME      SEDBSE
END
```

Example 3. Superelement Solution Phase

```
PROBLEM TITLE Superelement solution phase
QPM4 ELEMENT TOPOLOGY
 FIRST  1  2  3  6  5
 INC    1  3  3  3  3  4
NODE COORDINATES
 FIRST  2  5  0
 INC    1  5  0  2
 INC    3  0  5  5
QPM4 GEOMETRIC PROPERTIES
```

```
 1   0.01  0.01  0.01  0.01
GEOMETRIC ASSIGNMENTS
 1  4  1  1
MATERIAL PROPERTIES
 2  210E9  0.3
MATERIAL ASSIGNMENTS
 1  4  1  2
FREEDOM TEMPLATE
 U  V  THZ
SUPPORT NODES
 2  3  1  F  R  R
 14 15 1  F  R  R
LOAD CASE
 CL
  3  15  12    10  0  10
  6  12   3    20
C
C  Open the superelement database SEDBSE in the external
file SEEXTF
SUPERELEMENT ASSIGN   SEDBSE      SEEXTF
C
C  Include the superelement SENAME (which is stored in
database SEDBSE) into the analysis.
SUPERELEMENT USE            SENAME      SEDBSE
C
C  Include other superelements into the complete
structure
C  Include user modal data USE1 into the analysis.
SUPERELEMENT MODAL_DATA USE1
C
C  Include other user modal superelements into the
complete structure
EIGENVALUE CONTROL
```

```
CONSTANTS  5  D  D  1
PLOT FILE
END
```

Example 4. Superelement Recovery Phase

```
RESTART READ  1  SEREST
C  Recover the superelement SENAME results for the most
recent analysis
C
C  Open the superelement database SEDBSE in the external
file SEEXTF
SUPERELEMENT ASSIGN   SEDBSE      SEEXTF
C
C  Recover the superelement results
SUPERELEMENT RECOVER  SENAME      SEDBSE
RESTART WRITE
PLOT FILE
END
```

Example 5. User Modal Neutral File

```
FREE FORMAT
25  0  0  1
User modal data for two beams with total of six freedoms
26  0  0  1
22-09-93:12:00:V11.0
27  0  0  11  3  2 6
U   V  THZ
11  1  2  3
14  1  2  3
0.145609E+02    0.162796E+02    0.141941E+03
0.158695E+03    357.302
399.476
```

```
use1.adp
use2.adp
use3.adp
use4.adp
use5.adp
use6.adp
99
```

Example 6. User Modal Results File

```
USE1 - Two beams with total of six freedoms (One of six
result files)
 10   14   0.127609   14   3
use1    GLOBAL DISPLACEMENTS
 MODE  =   1
1   0.0000000E+00    0.0000000E+0     0.0000000E+00
2   0.0000000E+00    0.8983746E-17    0.2475554E-17
4   0.0000000E+00    0.0000000E+00    0.0000000E+00
5   0.0000000E+00    0.5336911E-16    0.1470636E-16
7   0.0000000E+00    0.0000000E+00    0.0000000E+00
8   0.0000000E+00    -.1740663E-15    -.4773746E-16
10  0.0000000E+00    0.0000000E+00    0.0000000E+00
11  0.0000000E+00    -.1007247E-14    -.2966510E-15
13  0.0000000E+00    0.0000000E+00    0.0000000E+00
14  0.0000000E+00    0.1276092E+00    0.3516390E-01
```

Analysis Control

The following types of analysis control data chapters are available for the control of specific LUSAS analysis types:

- NONLINEAR CONTROL — for nonlinear analysis
- DYNAMIC CONTROL — for step-by-step dynamic analysis
- TRANSIENT CONTROL — for transient field analysis
- VISCOUS CONTROL — for creep analysis
- EIGENVALUE CONTROL — for eigenvalue extraction analysis

- GUYAN CONTROL for Guyan reduced eigenvalue analysis
- MODAL_DAMPING CONTROL for computation of distributed damping
- SPECTRAL CONTROL for spectral response analysis
- HARMONIC_RESPONSE CONTROL for forced vibration analysis
- FOURIER CONTROL for Fourier analysis

Nonlinear Control

The NONLINEAR CONTROL data chapter is used to control the solution procedure for nonlinear analyses. For further information regarding the solution of nonlinear problems refer to Nonlinear Analysis, and the *LUSAS Theory Manual*.

```
NONLINEAR CONTROL
```

Incrementation

The INCREMENTATION data section specifies how an automatic solution is to proceed. Input only if using an automatic procedure.

- **Manual load incrementation** Manual incrementation may be specified by repetition of the LOAD CASE data chapter after the NONLINEAR CONTROL chapter, hence explicitly defining the required loading parameters. The general form of manual incrementation is therefore:

  ```
  LOAD CASE
  NONLINEAR CONTROL
  LOAD CASE
  LOAD CASE
  LOAD CASE etc.
  ```

  Where loads are being specified, the total values at each step must be input. In contrast, displacement increments are additive, and hence the incremental change should be specified. It should be noted that, when using manual incrementation, since the incrementation and final levels are defined explicitly, the INCREMENTATION and TERMINATION sections of the NONLINEAR CONTROL data chapter are redundant.

- **Automatic load incrementation** Automatic incrementation for nonlinear problems is controlled via the INCREMENTATION section of the NONLINEAR CONTROL data chapter. In this case, only the initial LOAD CASE is specified and the incrementation is controlled by the INCREMENTATION and TERMINATION sections of the NONLINEAR CONTROL data chapter. The general form of automatic incrementation is therefore:

  ```
  LOAD CASE
  NONLINEAR CONTROL
  ```

INCREMENTATION

TERMINATION

- When using automatic incrementation, the initial loading components specified in the LOAD CASE data chapter are multiplied by the current load factor. The starting load factor is specified as the parameter **slamda** in the INCREMENTATION data section.

INCREMENTATION slamda [dlamdx isurfc itd cstifs dellst delsmx]

slamda Starting reference load factor (default = 1.0 on first increment, default = 0.0 on subsequent increments).

This is the factor by which the reference load level will be multiplied for the predictor, or first iteration, of the next load increment. This load level will remain constant if **isurfc**=0 (constant load level). Must be nonzero at the first increment but may subsequently be respecified as zero if the new load factor is to be computed from the previous convergence history (i.e. using **itd**).

dlamdx The maximum absolute change in the load factor on iteration zero of any load increment.

This parameter is used when the load level is automatically adjusted following consideration of the iterations desired for convergence against those achieved. The effect of the parameter is to limit the rate of change of load in an increment. If zero is input, no limit will be applied. If a **D** is input, the default value of **dlamdx** = 2.0*(**tlamda** - **slamda**) is taken; **tlamda** is the total load factor at the end of the previous increment.

Note. If a nonzero value of **dellst** is specified in an analysis controlled by the arc-length method (**isurfc**=1 or 2) then:

- the maximum incremental arc-length parameter **delsmx** will be used to limit the step size of subsequent increments, and
- **slamda** will have no effect (i.e. **dellst** will be used to control the new increment).

isurfc The constant load level or arc-length control parameter (default=0)

If specified as zero, the loading will remain constant during the iteration process (that is, the 'constant load level procedure'). If input as one or two, the loading will vary during the iterations (that is, an 'arc-length' procedure, see *Notes* on TERMINATION). Two algorithms based on the arc-length method are available in LUSAS:

- Crisfield's modified approach
- Rheinboldt's arc-length method

itd The number of desired iterations per load increment (default = 4)

When using automatic incrementation, the loading variable (load or arc-length) is varied according to the number of iterations taken to converge on the preceding step. If the number of iterations taken for convergence exceeded the desired number, specified by **itd**, then the loading variable is decreased; conversely, if the number of iterations taken to converge was less than the desired number, the step length is increased. Hence the rate of change of loading variable is adjusted depending on the degree of nonlinearity present. If zero is input, the load variable will remain constant.

cstifs The current stiffness parameter value at which the solution will switch from a constant load level to an arc-length procedure (default = 0.4). The current stiffness parameter varies between 1.0 (initially) and 0.0 at a horizontal limit point. It is therefore a useful measure of structural collapse. One may wish to start with load control (**isurfc**=0) and have the program automatically switch to arc-length control as structural collapse is neared. Measuring structural stiffness with the current stiffness parameter, once the current stiffness parameter falls below the threshold value of **cstifs** the program automatically switches to arc-length control. If zero is input, the parameter is ignored.

Note. **itd** must be given a positive value to use this facility, and **isurfc** must equal zero.

dellst The incremental-length value required to restart an analysis under arc-length control. When applying a restart with the structure near to collapse, it is advisable to use arc-length control (**isurfc**=1). If the load-deflection response is very flat, it is impractical to restart (as normal) by specifying a load factor, **slamda**. Instead, it is better to specify an 'arc-length increment', **dellst**. An appropriate value for **dellst** can be obtained by looking at output values of the arc length in the iterative log or output files, of prior arc-length increments, **deltl**. If **dellst** is nonzero this value will be used for the new load increment instead of **slamda** no matter what value of **slamda** is specified.

delsmx The maximum value of **dellst** for subsequent increments (default is calculated as **delsmx** =2*__dellst__).

If **delsmx** is nonzero this value will be used to limit the size of subsequent increments instead of **dlamdx** no matter what value of **dlamdx** is specified.

Step Reduction

The STEP_REDUCTION data section defines how a load increment will be reduced if convergence difficulties occur. This section is only applicable when values have been specified under INCREMENTATION.

STEP_REDUCTION [mxstrd stpred stpfnl]

mxstrd The maximum number of times a step reduction can occur on a single increment (default = 5). If input as zero step reduction will be suppressed.

stpred The factor used to reduce the load increment on a step reduction (default = 0.5).

stpfnl The factor used to increase the original load increment if **mxstrd** step reductions have failed to achieve a solution (default = 2.0).

If **mxstrd** step reductions have failed to lead to convergence then a final attempt is made to achieve a solution by increasing the original increment using **stpfnl**. This procedure has the potential to 'step over' a difficult point in an analysis (e.g. a bifurcation point) so that the solution can continue.

Iterations

ITERATIONS specifies how the iteration strategy is to proceed.

```
ITERATIONS nit [nalps toline ampmx etmxa etmna isilcp]
itype
```

nit The maximum number of iterations for each load increment (default = 12).

nalps The maximum number of line searches to be carried out on each iteration (default = 2)

toline The line search tolerance factor (default = 0.75).

ampmx The maximum amplification factor in a line search (default = 5.0).

etmxa The maximum step length in a line search (default = 25.0).

etmna The minimum step length in a line search (default = 0.0).

isilcp Separate iterative loop for contact procedure

No separate iterative loop for contact = 0 (default)

Separate iterative loop for contact = 1

Line searches are carried out if the absolute value of **epsln** on the nonlinear iterative log exceeds the value **toline**. The aim is to make **epsln** reasonably small on each iteration to speed convergence and prevent divergence.

itype The type of iteration to be carried out for each iteration (default = NR)

Each iteration is specified on a separate line. There must be **nit** + 1 iteration types specified to include iteration zero. The type may be NR for Newton-Raphson or MNR for modified Newton-Raphson iterations. (If not included, the default is NR for all iterations). Note that FIRST, INC, INC can be used to generate this input.

For example, to apply 'standard modified NR' with the calculation of the stiffness matrix at the beginning of each increment the first iteration must be specified as NR.

Notes

- If arc-length is to be used, it is advisable to ensure that the stiffness is calculated at least at the beginning of the increment.

- To specify the separate contact iterative loop procedure the variable **isilcp** on the ITERATIONS data chapter should be set to one. This will invoke the procedure for relevant material models. Since the procedure is designed to deal with contact and nonlinear material interaction it only applies to those elements that can be used with slidelines. Other elements not attached to the slideline are dealt with in the normal manner.

Bracketing

BRACKETING can be used to locate a limit or bifurcation point during a geometrically nonlinear analysis (omit this command if bracketing is not required). When using the BRACKETING command, NONLINEAR CONTROL should be followed by EIGENVALUE CONTROL STIFFNESS. After the critical point in the nonlinear analysis has been found, the eigenvalue analysis is invoked. The scalar product of the lowest eigenmode extracted from the current tangent stiffness matrix and the applied load vector will indicate whether a limit or bifurcation point has been found. It is only possible to bracket the first critical point encountered in an analysis.

```
BRACKETING  [ibrac irevsb brtol]
```

ibrac Method used to locate limit/bifurcation point:
1 bi-section (default)
2 interpolation
3 Riks semi-direct approach

irevsb Reversible (elastic) or irreversible (plastic or path dependent) analysis:
0 reversible (default)
1 irreversible

brtol Bracketing tolerance (default = 0.01).

Depending on the type of arc-length solution selected, the bracketing tolerance can depend upon:

- the ratio of the current and initial minimum pivots and the ratio of the total and incremental load factors, or
- the ratio of the arc-length prior to bracketing and the subsequent increment in arc-length

The value specified for **brtol** should be small enough to accurately locate the critical point; too tight a tolerance will eventually lead to ill conditioning of the tangent stiffness. Trial solutions using various values for **brtol** will inevitably be required to obtain a solution of sufficient accuracy.

Branching

BRANCHING can be used to 'branch switch' onto a secondary path after a bifurcation point has been located using the BRACKETING command (omit this command if branching is not required). This procedure forces the analysis to leave the stable equilibrium path and follow an unstable secondary path. Two options exist for guiding the solution on to the secondary path:

- Eigenvalue injection
- Artificial force with Rheinboldt's arc-length

This facility should only be used by restarting a successfully completed bracketing analysis using the RESTART data chapter.

```
BRANCHING  [ibrnch dellst]
```

ibrnch Method used to branch on to secondary path:
1 eigenmode injection; used with **isurfc**=1 (default)
2 artificial force and Rheinboldt's arc-length; used with (**isurfc**=2)

dellst Starting arc-length scaling factor (default=0.01)

If the BRANCHING command is specified the appropriate value of **isurfc**, the type of solution algorithm to be adopted, will be set automatically to suit the branching method selected.

Note. Values assigned for **isurfc** and **dellst** under the BRANCHING command will override those specified using INCREMENTATION. Any previous value specified for **dellst** will also be overwritten with the default value if the **dellst** data location in BRANCHING is left blank.

Convergence

The CONVERGENCE data section specifies at which stage the iterative corrections can be assumed to have restored the structure to equilibrium. It is compulsory on the first occurrence of NONLINEAR CONTROL. The solution has converged if the values of **all** the following criteria computed after an iteration are less than those specified. If a parameter is input as zero, the corresponding criteria is ignored.

```
CONVERGENCE  rmaxal [rnoral dlnorm rlnorm wlnorm
dtnrml]
```

rmaxal The limit for the maximum absolute value of any residual (**mar**) (default = a large number).

rnoral The limit for the square root of the mean value of the squares of all residuals (**rms**) (default = a large number).

dlnorm The limit for the sum of the squares of the iterative displacements as a percentage of the sum of the squares of the total displacements (**dpnrm**). Only translational degrees of freedom are considered by default but all degrees of freedom can be included by specifying Option 187. (default = 1.0)

rlnorm The limit for the sum of the squares of all residual forces as a percentage of the sum of the squares of all external forces, including reactions (**rdnrm**). Only translational degrees of freedom are considered but all degrees of freedom can be included by specifying Option 187. (default = 0.1)

wlnorm The limit for the work done by the residuals acting through the iterative displacements as a percentage of the work done by the loads on iteration zero of the increment (**wdnrm**) (default = a large number).

dtnrml The limit for the sum of the squares of the iterative displacements as a percentage of the sum of the squares of the incremental displacements (**dtnrm**). Only translational degrees of freedom are considered by default but all degrees of freedom can be included by specifying Option 187. (default = 1.0)

Output

The OUTPUT data section specifies how often output is required. Insert zero to ignore a particular output option. On the last increment a PLOT FILE will automatically be written, overriding any specification in the OUTPUT command. This is also the case for output to the LUSAS output file and the LUSAS log file.

```
OUTPUT  nitout [incout incplt incrst nrstsv inclog
inchis]
```

nitout The iteration interval for output of results (default =0).

incout The increment interval for output of results (default = 1).

incplt The increment interval for writing of plotting data to the plot file (default = 1). PLOT FILE does not need to be specified.

incrst The increment interval for writing of problem data to the restart file (default = 0). RESTART WRITE does not need to be specified.

nrstsv The maximum number of restart dumps to be saved (default = 0). (to save the latest 2 dumps throughout the problem, set **nrstsv** = 2).

inclog The increment interval for writing the iterative log (default=1).

inchis The increment interval for writing the selective results history file. (default=1 and will only be invoked if selective results output is specified). In problems where the restart facility is used, a separate history file is created for each analysis.

Incremental Couple Read

The INCREMENTAL COUPLE READ data section controls the frequency of reading from the coupled datafile for thermo-mechanically coupled analyses (omit command if not performing a coupled analysis). See Thermo-Mechanically Coupled Analyses for more details.

```
INCREMENTAL COUPLE READ  [nffrd dtfrd niterr]
```

nfrrd The number of steps between data reads (default = 1)

dtfrd The increment of time between data reads (default = 0)

niterr The number of iterations between data reads (default = 0)

Incremental Couple Write

The INCREMENTAL COUPLE WRITE data section controls the frequency of writing to the coupled datafile for thermo-mechanically coupled analyses (omit command if not performing a coupled analysis). See Thermo-Mechanically Coupled Analyses for more details.

```
INCREMENTAL COUPLE WRITE  [nfrwt dtfwrt ndsave niterw]
```

nfrwt The number of steps between writes (default = 1)
dtfwrt The time increment between writes (default = 0)
ndsave The number of datasets retained on disk. It is recommended that at least two datasets are saved. (default = all)
niterw The number of iterations between writes (default = 0)

Termination

The TERMINATION data section specifies when an automatic incrementation solution is to terminate. It is compulsory if INCREMENTATION has been specified. If more than one termination criterion has been specified, termination will occur following the first criterion to be satisfied. Insert zero to ignore a particular option.

```
TERMINATION  tlamdx [maxinc maxnod mxvar rmxdsp]
```

tlamdx The maximum total load factor to be applied (default = 0)
maxinc The maximum number of further increments to be applied from the time of specification of this NONLINEAR CONTROL (default = 1).
mxnod The node number of displacement to be limited (default = 0).
mxvar The variable number of displacement to be limited at node **mxnod** (default = 0).
rmxdsp The maximum displacement of node **mxnod** variable **mxvar** (default = 0).

When automatic incrementation has been specified the analysis will terminate exactly at the specified value of **tlamdx** for analyses that are controlled by non arc-length automatic incrementation (i.e. **isurfc** = 0).

Notes

1. The INCREMENTATION and TERMINATION data sections are only used in conjunction with automatic load incrementation.
2. The INCREMENTAL COUPLE READ and INCREMENTAL COUPLE WRITE data sections are only used in thermo-mechanically coupled analyses.
3. Automatic load step reduction will not take place in coupled analyses.
4. Throughout NONLINEAR CONTROL, where stated, you may input a D for a particular input variable to obtain the default value.
5. When performing a nonlinear transient, dynamic or creep analysis, the NONLINEAR CONTROL chapter must be used to specify ITERATION and

CONVERGENCE only. Parameters controlling the number of problem steps, incrementation of loading etc. must be set using the commands in the TRANSIENT, DYNAMIC or VISCOUS CONTROL chapters.

6. When carrying out a nonlinear analysis the plot and restart dumps should be requested by specifying the appropriate parameters in the OUTPUT data section. However, a PLOT or a RESTART command may appear after this data chapter.
7. The nonlinear solution may be continued after the solution has failed to converge at a particular increment by setting Option 16.
8. TERMINATION must be used in conjunction with load curves to set the desired number of load increments.
9. When using an arc-length procedure, Option 164 forces the arc-length solution to be guided by the current stiffness parameter, `cstifs`, instead of using the minimum pivot, `pivmn`. If a bifurcation point is encountered, the arc-length procedure could cause the solution to oscillate about this point with no further progress being made. Option 164 allows the solution to continue on the fundamental path and overcomes any such oscillations if a bifurcation point is encountered. Note that Option 164 is not valid when using the BRACKETING command.
10. BRANCHING and BRACKETING must not be specified if they are not required in an analysis.
11. STEP_REDUCTION is only applicable if values have been specified under the INCREMENTATION data section.
12. Specifying Option 62 forces the solution to continue if more than two pivots are encountered during an analysis.
13. Option 131 is switched off by default for a nonlinear analysis, but this may be set to on in the data file if required. If Option 131 is set to on a warning message is issued and the analysis will continue with the option switched on.
14. The following nonlinear parameters are output:

MAR	The maximum absolute residual
RMS	The root mean square of the residuals
DPNRM	The displacement norm as a percentage of the total displacements
RDNRM	The residual force norm as a percentage of the total reactions
WDNRM	The work done by the residual forces as a percentage of the work done by the loads on iteration zero
DTNRM	The displacement norm as a percentage of the total displacements for the increment
EPSLN	The line search tolerance parameter
ETA	The final line search step length
DELTL	The incremental displacement length (arc-length)
DELTW	The work done by the external loads during the current increment
DLMDA	The change in load factor on the current iteration

TLMDA The total load factor
LTDSP The value of the displacement at the node and variable number specified in the TERMINATION criteria
MXSTP The maximum number of plasticity steps used (if MXSTP ≥ 1 plasticity has occurred in that iteration)
NLSCH The number of line searches carried out on the current iteration
CSTIF The current stiffness parameter
PENMX The maximum penetration on slidelines
NDPMX The slideline node at which maximum penetration occurs
KDSMX The node: variable number at which the maximum incremental displacement occurs
ISURF The load/arc-length control parameter
ENGY Total strain energy
PLWRK Total plastic work
PIVMN The minimum pivot value from the tangent stiffness
PIVMX The maximum pivot value from the tangent stiffness
NSCH The number of negative pivots found during the solution
KPVMN The node: variable number at which the minimum pivot occurs
KPVMX The node: variable number at which the maximum pivot occurs

Slideline

The augmented Lagrangian method with slidelines employs both penalty parameters (contact stiffnesses) and Lagrangian multipliers to reduce contact penetrations. At convergence, the method checks the penetrations against a tolerance. If any are outside the tolerance the Lagrangian multipliers, and if the penalty update procedure is specified, the contact stiffnesses are updated at the relevant nodes and the solution is rerun.

The progress of the solution is controlled by the SLIDELINE data section, which places a limit on the number of Lagrangian multiplier and penalty parameter updates.

```
SLIDELINE NAGLMX NPUPMX
```

Naglmx Maximum number of augmented Lagrangian updates per increment (default = 2).
Npupmx Maximum number of penalty parameter (contact stiffness) updates per increment (default = 1).

Application of Loads with Nonlinear Control

The stress state is always 'remembered' on iteration zero after a change of control. In a nonlinear analysis the stresses will be updated based on the external loads until convergence is achieved. If the load is reduced to zero in a subsequent loadcase, then the resulting stresses may also reduce to zero while any plastic strains will remain.

The type of loading is important when determining whether a load level from one increment is "remembered" or " forgotten" in subsequent increments when using manual or automatic load incrementation (see also *General Loading Rules*). Some loading types are total in nature whilst other are incremental. All of the loading types in LUSAS are total with the exception of the Incremental Prescribed Displacement loading type (PDSP)

- **Total**

  These apply **only** the specified loading magnitude to the structure. This means that any previously specified loading at a node or element is overwritten by the current specification of load.

  Consider an automatic, fixed load increment in which a concentrated load magnitude of 5.0 is to be incremented from a reference load factor of 0.2 over 3 increments. The initial load applied will be calculated according to 0.2*5.0. The subsequent two load increments will be calculated as 0.4*5.0 and 0.6*5.0

- **Incremental**

  The specified loading magnitude is applied **in addition** to the load already applied to the structure.

  Consider the example above but with incremental prescribed displacement loading. The initial load applied will be calculated according to 0.2*5.0. The subsequent two load increments will be calculated incrementally as (previous load + next increment of load), i.e.

  2nd Increment: 0.2*5.0 + (0.4 - 0.2)*5.0

  3rd Increment: 0.4*5.0 + (0.6 - 0.4)*5.0

The following examples illustrate different orders in the command formats for NONLINEAR CONTROL. These examples are not affected by the presence of:

- Material nonlinearity
- Geometric nonlinearity
- Arc-length procedures
- Line search procedures
- Dynamic analyses
- Static analyses

Note. Automatic nonlinear control is not applicable for use with a transient dynamic analysis.

Application Of Force Loading

Force loading is any loading EXCEPT incremental or total prescribed displacement loading.

Example. Automatic Incrementation Followed by Automatic

```
CL
 1  0  0          10.0
NONLINEAR CONTROL
 INCREMENTATION   0.5
 TERMINATION      0    3
```

Giving a load at the end of this control of 15.

```
CL
 1  0  0          20.0
NONLINEAR CONTROL
 INCREMENTATION   1.0
 TERMINATION      0    3
```

Giving a load at the end of this control of 60.

The load is applied in steps 5, 10, 15, 20, 40, 60. In this sense it could be said that the previous load is 'forgotten'.

Example. Manual Incrementation Followed by Manual

```
CL
 1  0  0          10.0
NONLINEAR CONTROL
 ITERATION        12
 CONVERGENCE      0    0    D    D
```

Giving a load at the end of this control of 10.

```
CL
 1  0  0          20.0
NONLINEAR CONTROL
 ITERATION        12
 CONVERGENCE      0    0    D    D
```

Giving a load at the end of this control of 20.

Example. Automatic Incrementation Followed by Manual

```
CL
 1  0  0          10.0
NONLINEAR CONTROL
 INCREMENTATION   0.5
 TERMINATION      0    3
```

Giving a load at the end of this control of 15

```
CL
 1  0  0          10.0
NONLINEAR CONTROL
 ITERATION        12
 CONVERGENCE      0    0    D    D
```

Giving a load at the end of this control of 10.

Example. Manual Incrementation Followed by Automatic

This is the **only** case where a load is **remembered**. The application of a manual increment will remain throughout the analysis until any further manual increment. Additional load is applied from the **current** automatic incrementation.

```
CL
 1  0  0          10.0
NONLINEAR CONTROL
 ITERATION        12
 CONVERGENCE      0    0    D    D
```

Giving a load at the end of this control of 10.

```
CL
 1  0  0          2.0
NONLINEAR CONTROL
 INCREMENTATION   1.0
 TERMINATION      0    3
```

Giving a load at the end of this control of 16 (10 + 6).

Dead Load and Live Load

A typical nonlinear analysis would require the application of a **dead** load, typically self-weight, as a constant load for the analysis, together with the **live** load which is gradually applied over a number of increments. The following examples show this can be achieved.

Example 1. Dead Load and Incremented Live Load

Using the characteristics of an automatic nonlinear loading following a manual load application:

```
CL
 1  0  0          10.0
NONLINEAR CONTROL
 ITERATION        12
 CONVERGENCE      0    0    D    D
```

Which, at the end of this control, applies a dead load of 10.

```
CL
 1  0  0          2.0
NONLINEAR CONTROL
 INCREMENTATION   1.0
 TERMINATION      0    3
```

Which, at the end of the second control, applies a dead load of 10 and a live load of 6 giving a total load of 16.

Manual Load Application

```
CL
 1  0  0          10.0
NONLINEAR CONTROL
 ITERATION        12
 CONVERGENCE      0    0    D    D
```

Which, at the end of this control, applies a dead load of 10.

```
CL
 1  0  0          10.0
 1  0  0          2.0
```

```
CL
 1  0  0          10.0
 1  0  0          4.0
CL
 1  0  0          10.0
 1  0  0          6.0
```

Which, at the end of the second control, applies a dead load of 10 and a live load of 6 giving a total of 16.

Using Load Curves

The load curve facility allows two load variations to be applied. One constant and one variable as required.

```
CL
 1  0  0     10.0
CL
 1  0  0     2.0
CURVE DEFINITION 1 COSINE
 1.0  0.0
CURVE DEFINITION 2 USER
 1   0.0
 4   3.0
CURVE ASSIGNMENT
 1   1.0  1
 2   1.0  2
NONLINEAR CONTROL
 TERMINATION    0    4
```

Which applies a constant load level of 10 and a varying live load of 6 giving a total load of 16.

Follower Forces

In general, follower forces are available in LUSAS when the loading type is either UDL (Uniformly Distributed Load) or FLD (Face Load) and Eulerian geometric nonlinearity option (Option 167) is selected. For certain elements additional follower loading is also available using Option 36 (see the *LUSAS Element Reference Manual*).

Explicit dynamics elements will also give follower forces when using the FLD load facility.

General Loading Rules

1. In switching from manual to automatic control, any loading input under the manual control is remembered and held constant while the automatic procedure is operating
2. In switching from automatic back to manual control, any loading accumulated under automatic control is forgotten and only the manual load is applied. To include the final load level from the automatic load increments, the load datasets from which it comprises must be assigned to this manual load case
3. In switching from manual to manual control, any loading accumulated under previous manual controls is forgotten and only the current manual load is applied.
4. In switching from automatic to automatic control, any loading accumulated under the previous automatic control is forgotten and only the current automatic load is applied. To include the final load level from the previous automatic load increment, an additional loadcase must be inserted prior to the next automatic increment and the load datasets from the previous automatic increment assigned to this manual load case.
5. If incremental prescribed displacements are being used then, in any switching from one type of control to another, the effect of prescribed displacements will be remembered and will not need to be input again. This is not the case for total prescribed displacements which are total loads
6. The stress and strain state will always be remembered whatever loading is applied on iteration zero after the change of control. In a nonlinear analysis the stresses/strains will be updated based on the external loads until convergence is achieved. If the load is reduced to zero in a subsequent loadcase, then the resulting stresses may also reduce to zero while any plastic strains will remain. Additionally, if the same load magnitude is maintained across a change of control, convergence will be achieved in one iteration because the stress state and the equilibrium internal forces do not change.

Dynamic Control

The DYNAMIC CONTROL data chapter is used to control the solution procedure for transient dynamic analyses. For further information regarding the solution of transient dynamics problems refer to Dynamic Analysis, and the *LUSAS Theory Manual*.

```
DYNAMIC CONTROL
```

Incrementation

The INCREMENTATION data section controls the applied time step interval.

```
INCREMENTATION  dt  [tsfac dtincf inctyp dtmin dtmax]
```

dt The initial time step.

tsfac The scale factor for computing the time step for explicit dynamic analyses (default = 0.9 for 3D and 2/3 for 2D)

dtincf The time step increment restriction factor for dynamic analyses (default = 1.0 for explicit analysis, default = 10.0 for implicit analysis). Note that stability conditions must be taken into account when setting **dtincf**. The default value may lead to instability in some cases.

inctyp Set to 1 for a driven coupled analysis. This means that the time step for this analysis is calculated by the thermal analysis. Note that a driven coupled analysis is not permitted when coupling iteratively. (default = 0).

dtmin Minimum time step for explicit dynamic analyses (default = 0.0). Note that stability conditions must be taken into account when setting **dtmin** since solution instability may occur.

dtmax Maximum time step for explicit dynamic analyses (default = no limit)

Constants

The CONSTANTS data section specifies the time integration factors to be used. If not specified, the default or previous user defined data are assumed. If explicit dynamic elements have been specified then the default constants will be those for an explicit dynamic analysis and this command may be omitted.

Implicit dynamics (Hilber-Hughes integration scheme)

```
CONSTANTS  alpha  [beta gamma]
```

alpha Integration factor (default = 0).

beta Integration factor (default = 1/4).

gamma Integration factor (default = 1/2).

Note. For an unconditionally stable, second order accurate scheme, **alpha** must lie between the limits:

$$-\frac{1}{3} \le a \le 0$$

and **beta** and **gamma** are defined as:

$$\beta = \frac{(1-\alpha)^2}{4}, \qquad \gamma = \frac{1-2\alpha}{2}$$

If **beta** and **gamma** are not specified they will be computed automatically using the above equations. Setting **alpha** to zero reduces the algorithm to the Newmark method (this is the default).

Explicit dynamics

alpha Integration factor = 0
beta Integration factor = 0
gamma Integration factor = ½

Note. For an explicit dynamic analysis the above integration factors are mandatory and the CONSTANTS data section may be omitted in this case. Only explicit dynamics elements may be used with the explicit dynamics constants.

Incremental Couple Read

The INCREMENTAL COUPLE READ data section controls the frequency of reading from a coupled datafile for a coupled analysis (omit if not performing a coupled analysis). See Coupled Analysis for more information.

```
INCREMENTAL COUPLE READ  [nffrd dtfrd niterr]
```

nfrrd The number of steps between data reads (default = 1)
dtfrd The increment of time between data reads (default = 0)
niterr The number of iterations between data reads (default = 0)

Incremental Couple Write

The INCREMENTAL COUPLE WRITE data section controls the frequency of writing a coupled datafile for a coupled analysis (omit if not performing a coupled analysis). See Coupled Analysis for more information.

```
INCREMENTAL COUPLE WRITE  [nfrwt dtfwrt ndsave niterw]
```

nfrwt The number of steps between writes (default = 1)
dtfwrt The time increment between writes (default = 0)
ndsave The number of datasets retained on disk (default = all)
niterw The number of iterations between writes (default = 0)

Output

The OUTPUT data section specifies how often output is required. If not specified, the default or previously defined user data are assumed. On the last increment a PLOT FILE will automatically be written, overriding any specification in the OUTPUT command. This is also the case for output to the LUSAS output file and the LUSAS log file.

```
OUTPUT  incout  [incplt incrst nrstsv inclog inchis]
```

incout Increment interval for output of results (default = 1)
incplt Increment interval for writing of plotting data to the plot file (default = 1). PLOT FILE does not need to be specified.
incrst The increment interval for writing of problem data to the restart file (default = 0). RESTART WRITE does not need to be specified.

nrstsv The maximum number of restart dumps to be saved (default =0). (to save the latest 2 dumps throughout the problem, set **nrstsv** = 2)

inclog The increment interval for writing the time step log (default=1).

inchis The increment interval for writing the selective results history file. (default=1 and will only be invoked if selective results output is specified). In problems where the restart facility is used, a separate history file is created for each analysis.

Termination

The TERMINATION data section specifies when the analysis terminates. If not specified, the default or previously defined user data are assumed.

```
TERMINATION  maxinc  [ttime dtterm]
```

maxinc The maximum number of time steps to be applied (default = 1).

ttime The total response time at which the analysis should terminate (default = a large number).

dtterm The minimum time step below which the analysis should terminate (default = 0.0)

Notes

1. To start a dynamic analysis, a knowledge of the initial conditions is required. For example, initial displacements may be computed from a static pre-analysis and initial velocities calculated. These velocities can then be specified by the user in the dynamic analysis using the VELOCITY data chapter.
2. For an analysis using explicit dynamic elements only **dt** is taken as the smaller of the user input and the calculated values (**dt** may be entered, in this instance, as zero). Further steps are automatically adjusted according to mesh deformation.
3. **dtterm** will terminate an analysis if the step size is reduced below this value.
4. If **inctyp** is set to 1 for a driven coupled analysis, `dt` is re-interpreted as the maximum step size permitted, and the new step size is calculated to be as close to, but not exceeding `dt`, as possible. The variable step size is always selected to ensure that the next data transfer takes place at the next dataset.
5. If both **maxinc** and **ttime** are specified, termination occurs with the minimum response time.
6. If the problem is nonlinear, the convergence and the iteration details must be set by the NONLINEAR CONTROL commands.
7. Automatic time stepping is mandatory for explicit dynamic analyses.
8. The INCREMENTAL COUPLE READ and INCREMENTAL COUPLE WRITE data sections should be omitted if a thermo-mechanically coupled analysis is not being performed.
9. Automatic load step reduction will not take place in a coupled analysis.

10. When carrying out a DYNAMIC analysis the plot and restart dumps should be requested by specifying the appropriate parameters in the OUTPUT data section. However, a PLOT or a RESTART command may appear after this data chapter.
11. For large explicit dynamic analyses the output files may be enormous if default values are chosen for output.
12. Throughout DYNAMIC CONTROL, where stated, you can input a D for a particular input variable to obtain the default value.
13. If no output or plot dump is requested, the stress computation is automatically bypassed during a linear dynamic analysis.

Example. Static Starting Conditions for Dynamic Analyses

In the following examples, non-essential data has been omitted.

Linear for the simple case of a constant dead load (simulated using **constant body force loading**), followed by a constant live load (using the **concentrated loading** type):

```
SUPPORT NODE
LOAD CASE
  CL
ELEMENT OUTPUT
  1 0 0 0
LOAD CASE
  CL
DYNAMIC CONTROL
END
```

For the more complex case of a constant dead load followed by a varying live load:

```
SUPPORT NODES
LOAD CASE
  CBF
ELEMENT OUTPUT
  1 0 0 0
LOAD CASE
  CBF
LOAD CASE
```

```
  CL
CURVE DEFINITION 1 USER
CURVE DEFINITION 2 USER
CURVE ASSIGNMENT
  1  1  1
  2  1  2
DYNAMIC CONTROL HILBER
END
```

Nonlinear Similarly, a constant dead load (simulated using **constant body force loading**), followed by a constant live load (using the **concentrated loading** type):

```
SUPPORT NODE
LOAD CASE
  CL
ELEMENT OUTPUT
  1 0 0 0
NONLINEAR CONTROL
LOAD CASE
  CL
DYNAMIC CONTROL
END
```

For a varying live load using load curves

```
SUPPORT NODES
LOAD CASE
  CBF
LOAD CASE
  PDSP 2
CURVE DEFINITION 1 COSINE
CURVE DEFINITION 2 COSINE
CURVE ASSIGNMENT
  1  1  1
```

```
ELEMENT OUTPUT
  1  0  0  0
NONLINEAR CONTROL
CURVE ASSIGNMENT
  1  1  1
  2  1  2
DYNAMIC CONTROL HILBER
END
```

Dynamic Integration Schemes

Integration Scheme	alpha	beta	gamma
Central Difference	-	0	½
Hilber-Hughes-Taylor	0	¼	½

The table shows the integration schemes available and the default integration parameters for each. The parameter values listed are set by default where the Central Difference scheme applies for explicit dynamic elements. The parameters are defined under the CONSTANTS data section of DYNAMIC CONTROL.

The constant **alpha** is used to control the amount of numerical damping within the Hilber-Hughes-Taylor solution scheme; it is not used in the explicit Central Difference scheme.

Selecting the Time Step

The time step for dynamic analysis is specified using the INCREMENTATION section of the DYNAMIC CONTROL data chapter. The selection of the time step is governed by stability and accuracy. When using explicit algorithms, stability requires that the time step is less than or equal to a critical value where:

$$t_{cr} = \frac{1}{\pi} T_{max} = \frac{2}{\omega_{max}}$$

where w_{max} and T_{max} are the circular frequency and period of the highest mode of vibration in the uncoupled system.

Transient Control

The TRANSIENT CONTROL data chapter is used to control the solution procedure for transient field analyses. For further information regarding the solution of transient

field problems refer to Transient and Dynamic Analyses, and the *LUSAS Theory Manual*.

```
TRANSIENT CONTROL
```

Incrementation

The INCREMENTATION data section controls the time step value for a transient field analysis.

```
INCREMENTATION  dt  [inctyp] dtincf dtmin dtmax
```

dt Initial time step value.

inctyp Set to 1 for a driven coupled analysis. This means that the time step for this analysis is calculated by the structural analysis. Note that a driven coupled analysis is not permitted when coupling iteratively (default = 0).

dtincf The time step increment factor (default=1 for constant time step)

dtmin The minimum permissible time step size (default=dt)

dtmax The maximum permissible time step size (default=dt)

Constants

The CONSTANTS data section is used to specify the time integration factor. If not specified, the default or previously defined user data are assumed. See Thermo-Mechanically Coupled Analyses for more details.

```
CONSTANTS  beta
```

beta The time integration factor
=1/2 for the Crank-Nicholson scheme
=1 for the backward difference scheme.
(default = 2/3 for linear analysis).
(default = 1 for nonlinear analysis, this value may not be overwritten).

Incremental Couple Read

The INCREMENTAL COUPLE READ data section is used to control reading from the coupled datafile for thermo-mechanically coupled analyses (omit command if not performing a coupled analysis). See Thermo-Mechanically Coupled Analyses for more details.

```
INCREMENTAL COUPLE READ  [nffrd dtfrd niterr]
```

nffrd The number of steps between data reads (default = 1)

dtfrd The increment of time between data reads (default = 0)

niterr The number of iterations between data reads (default = 0)

Incremental Couple Write

The INCREMENTAL COUPLE WRITE data section is used to control writing to the coupled datafile for thermo-mechanically coupled analyses (omit command if not performing a coupled analysis). See Thermo-Mechanically Coupled Analyses for more details.

```
INCREMENTAL COUPLE WRITE  [nfrwt dtfwrt ndsave niterw]
```

nfrwt The number of steps between writes (default = 1)
dtfwrt The time increment between writes (default = 0)
ndsave The number of datasets retained on disk (default = all)
niterw The number of iterations between writes (default = 0)

Output

The output frequency control. If not specified, the default or previously defined user data are assumed. On the last increment a PLOT FILE will automatically be written, overriding any specification in the OUTPUT command. This is also the case for output to the output file and the log file.

```
OUTPUT  incout  [incplt incrst nrstsv inclog inchis]
```

incout The increment interval for output of results (default = 1)
incplt The increment interval for writing of plotting data to the plot file (default = 1). PLOT FILE does not need to be specified.
incrst The increment interval for writing of problem data to the restart file (default = 0). RESTART WRITE does not need to be specified.
nrstsv The maximum number of restart dumps to be saved (default =0). (to save the latest 2 dumps throughout the problem, set **nrstsv**= 2)
inclog The increment interval for writing the time step log (default=1).
inchis The increment interval for writing the selective results history file. (default=1 and will only be invoked if selective results output is specified). In problems where the restart facility is used, a separate history file is created for each analysis.

Termination

The TERMINATION data section is used to specify when the analysis terminates. If not specified, the default or previously defined user data are assumed.

```
TERMINATION  maxinc  [ttime]
```

maxinc The maximum number of time steps to be applied (default = 1).
ttime The total response time at which the analysis should terminate (default = a large number).

Notes

1. Transient analysis is applicable to field problems only.
2. The INCREMENTAL COUPLE READ and INCREMENTAL COUPLE WRITE data sections should be omitted if the problem is not a thermo-mechanical coupled analysis.
3. Automatic load step reduction will not take place in a coupled analysis.
4. The support conditions and loading data specified prior to the first TRANSIENT CONTROL chapter provide the static steady state solution at time equals zero.
5. For a coupled analysis **only**, if inctyp is set to a nonzero value for a driven coupled analysis, dt is re-interpreted as the maximum step wise permitted, and the new step wise is calculated to be as close to, but not exceeding dt. The variable step wise is always selected to ensure that the next data transfer takes place exactly at the next dataset.
6. If both maxinc and ttime are specified, termination occurs with the minimum response time.
7. If the problem is nonlinear, the convergence and the iteration details must be set by the NONLINEAR CONTROL commands.
8. When carrying out a TRANSIENT analysis the plot and restart dumps should be requested by specifying the appropriate parameters in the OUTPUT data section. However, a PLOT or a RESTART command may appear after this data chapter.
9. Throughout TRANSIENT CONTROL, where stated, you can input a D for a particular input variable to obtain the default value.
10. Oscillatory response may be seen when using the default beta constant with step type loading. Increasing beta will reduce these oscillations (as would a more gradual ramping of the load).

Viscous Control

The VISCOUS CONTROL data chapter is used to control the solution procedure for creep analyses. For further information regarding the solution of problems involving rate dependent material nonlinearity refer to the *LUSAS Theory Manual*.

```
VISCOUS CONTROL
```

Incrementation

The INCREMENTATION data section controls the applied time step interval.

```
INCREMENTATION  [dt dtincf inctyp dtmin dtmax exptim]
```

dt	Time step (default = 0.001)
dtincf	Time step increment restriction factor (default = 10.0).
inctyp	Set to 1 for a driven coupled analysis. This means that the time step for this analysis is calculated by the thermal analysis. Note that a driven coupled analysis is not permitted when coupling iteratively (default = 0).
dtmin	Minimum time step (default = 0.0)
dtmax	Maximum time step (default = no limit)

exptim Exponent used to increase time step when using the CEB-FIP creep model (see Notes)

Automatic Viscous Control

The AUTOMATIC VISCOUS_CONTROL data section specifies the automatic time-stepping criterion to be applied and the factors to be used for each criterion.

```
AUTOMATIC VISCOUS_CONTROL  [OFF]
ictol  toler  control factor
```

ictol Control criterion (see below).

Toler Tolerance factor for controlling the automatic time step (see below). The way in which this factor is used is dependent upon the chosen control criterion.

Control factor May have one of two meanings depending on the criterion used (see *Notes* below):

- Permitted difference between the time step evaluated at the beginning of a step and that calculated once values at **t+dt** are established. If the error tolerance is exceeded the time step is halved.
- Switch for explicit time step calculation.

Notes

Definition of automatic time stepping criteria:

1. **ictol** = 1

toler = **dcrpmx** = incremental creep strain

control factor = Permitted difference between the time step evaluated at the beginning of a step and that calculated once values at **t+dt** are established.

Limit of change of creep strain rate dec/dt calculated at time **t** and **t+dt**:

$$d_t = \frac{\text{dcrpmx}}{\dot{\varepsilon}^c_{t+dt} - \dot{\varepsilon}^c_t}$$

2. If the gradient is almost constant, then a large **dt** will be predicted; conversely, if the difference in gradients is large a small time step will be predicted.

ictol = 2

toler = restriction factor

control factor = 0 (de-activate explicit step calculation)
1 (activate explicit step calculation)

The explicit time step is calculated using the creep strain rate **dec/dt**:

$$dt = \frac{4(1+\nu)}{3E\dfrac{d\dot{\varepsilon}^c}{dq}} \text{ x restriction factor}$$

where **E** is the Elastic modulus and **n** is Poisson's ratio of the material and **q** is the equivalent stress.

3. **ictol** = 3

toler = scale factor a

control factor = Permitted difference between the time step evaluated at the beginning of a step and that calculated once values at t+dt are established.

dt is calculated as a function of the total strain **ε** and the creep strain rate **dec/dt**:

$$dt = \left[\frac{\varepsilon \cdot \varepsilon}{\dot{\varepsilon}^c \cdot \dot{\varepsilon}^c}\right]^{\frac{1}{2}}$$

4. **ictol** = 4

toler = scale factor a

control factor = Permitted difference between the time step evaluated at the beginning of a step and that calculated once values at t+dt are established.

5. **dt** is calculated as a function of the critical total elastic strain components **εie** and the critical creep strain rate **deic/dt** component:

$$dt = \alpha \min\left|\frac{\varepsilon_i^e}{\dot{\varepsilon}_i^c}\right| \quad i=1, \text{ number of strain components}$$

Typical values of a are as following:

0.01< a<0.15	explicit analysis
a<10	implicit analysis

Default values are:

ictol	**toler**	**control factor**	
1	0.001	0.1	
2	1.000	1.0	(initialised for first step of implicit analysis then de-activated)
3	0.000	0.0	
4	0.000	0.0	

Rate Dependent

The RATE_DEPENDENT data section specifies whether implicit or explicit integration is to be used. If this section is omitted the default of implicit integration is invoked.

```
RATE_DEPENDENT {IMPLICIT | EXPLICIT}
```

State

The STATE data section can be used with the RESIN CURE MODEL, which is used for predicting the deformations of thermoset composites that occur during a hot cure manufacturing process. The use of the STATE data section simplifies the analysis so that only a structural analysis is required. This approach is restricted to thin parts in which the state of cure is constant, or nearly constant at every point. A globally defined state is then valid which can be evaluated from a basic knowledge of the cure cycle and cure kinetics. The cure is modelled by splitting it into sections in which a single material state is valid. As the material is linear within each state the shrinkage and thermal strains need only be considered at the end of the step.

```
STATE  < iState_i >_i=1,maxinc
```

iState The state of the resin, (liquid = 1, rubbery = 2, glassy = 3).
maxinc The maximum number of time steps to be applied.

This procedure requires a pseudo time step of 1, in other words, using

```
INCR 1
```

each iState is associated with a pseudo time multiple of 1 which is used to synchronise both the shrinkage and thermal data corresponding to the step, see *Notes*.

Incremental Couple Read

The INCREMENTAL COUPLE READ data section controls the frequency of reading from a coupled datafile for a coupled analysis (omit if not performing a coupled analysis).

```
INCREMENTAL COUPLE READ  [nffrd dtfrd niterr]
```

nfrrd The number of steps between data reads (default = 1)
dtfrd The increment of time between data reads (default = 0)
niterr The number of iterations between data reads (default = 0)

Incremental Couple Write

The INCREMENTAL COUPLE WRITE data section controls the frequency of writing a coupled datafile for a coupled analysis (omit if not performing a coupled analysis).

```
INCREMENTAL COUPLE WRITE  [nfrwt dtfwrt ndsave niterw]
```

nfrwt The number of steps between writes (default = 1)
dtfwrt The time increment between writes (default = 0)
ndsave The number of datasets retained on disk (default = all)
niterw The number of iterations between writes (default = 0)

Output

The OUTPUT data section specifies how often output is required. If not specified, the default or previously defined user data are assumed. On the last increment a PLOT FILE will automatically be written, overriding any specification in the OUTPUT command. This is also the case for output to the LUSAS output file and the LUSAS log file.

```
OUTPUT  [incout incplt incrst nrstsv inclog inchis]
```

incout Increment interval for output of results (default = 1)
incplt Increment interval for writing of plotting data to the plot file (default = 1). PLOT FILE does not need to be specified.
incrst The increment interval for writing of problem data to the restart file (default = 0). RESTART WRITE does not need to be specified.
nrstsv The maximum number of restart dumps to be saved (default = 0). For example, to save the latest 2 dumps throughout the problem, set **nrstsv** = 2.
inclog The increment interval for writing the time step log (default = 1).
inchis The increment interval for writing the selective results history file. (default=1 and will only be invoked if selective results output is specified). In problems where the restart facility is used, a separate history file is created for each analysis.

Termination

The TERMINATION data section specifies when the analysis terminates. If not specified, the default or previously defined user data are assumed.

```
TERMINATION  [maxinc ttime dtterm steady]
```

maxinc The maximum number of time steps to be applied (default = 1).
ttime The total response time at which the analysis should terminate (default = a large number).
dtterm The minimum time step below which the analysis should terminate (default = 0.0).
steady Value to terminate analysis relating the current displacement increment to the displacement increment directly following a new control data section.

$$\frac{\text{incremental displacement on current step}}{\text{incremental displacement norm on first step}} < \text{steady}$$

Notes

1. The starting time step for both explicit and implicit analyses is automatically calculated using the explicit step stability criterion. On subsequent steps, for implicit analyses, the step size is determined by the active automatic control criterion. The other active criteria are also evaluated on the first step and if any prove more critical than the explicit time step, then this value will be applied.
2. For many functions the creep rate is infinite at t=0 and therefore dt/2 is used to evaluate the creep strain rate and various derivatives required in the automatic step evaluation. The choice of dt determines the particular gradients and hence the calculation of the initial time step.
3. dtincf limits the growth of the time step. The default value allows the step to grow an order of magnitude with each time step, thus if the explicit time step is too conservative for the implicit analysis it does not take many time steps to adjust.
4. For an analysis using explicit dynamic elements, explicit integration must be specified.
5. If inctyp is set to a nonzero value for a driven coupled analysis, dt is re-interpreted as the maximum step wise permitted, and the new step wise is calculated to be as close to, but not exceeding dt, as possible. The variable step wise is always selected to ensure that the next data transfer takes place at the next dataset.
6. If both maxinc and ttime are specified, termination occurs with the minimum response time.
7. If the problem is nonlinear, the convergence and the iteration details must be set by the NONLINEAR CONTROL commands.
8. The INCREMENTAL COUPLE READ and INCREMENTAL COUPLE WRITE data sections should be omitted if a thermo-mechanically coupled analysis is not being performed.
9. Automatic load step reduction will not take place in a coupled analysis. When carrying out a VISCOUS analysis the plot and restart dumps should be requested by specifying the appropriate parameters in the OUTPUT data section. However, a PLOT or a RESTART command may appear after this data chapter.
10. Throughout VISCOUS CONTROL, where stated, you can input a D for a particular input variable to obtain the default value.
11. DYNAMIC CONTROL may be utilised with VISCOUS CONTROL if required.
12. NONLINEAR CONTROL must be specified with VISCOUS CONTROL unless linear materials are used with an explicit creep integration scheme (i.e. rate-dependent = explicit).
13. An example of using the STATE command to control resin shrinkage and thermal strains is given below.

```
SHRINKAGE PROPERTIES GENERAL TIME 3
ishr  …
          0        -0.035
          1        -0.045
          2        -0.046
:
LOAD CASE 1                                          TEMP
```

```
1                                                    First  Last Inc
1.0                                                  :
                                                     CURVE DEFINITION
1 USER                                                    0      0
                                                          1      0
                                                          2     -160
                                                     :
                                                     VISCOUS CONTROL
                                                STATE 2 3 3
```

The preceding data results in three load steps. On the first the composite is in its rubbery state (=2) and the shrinkage is 3.5%. Note that we are commencing the analysis when the resin is in its rubbery state at a temperature of 180°C; the load curve defines the cure temperature changes. On the second step the composite will have vitrified (glassy = 3) and a further 1% shrinkage will occur. Finally on step 3, the composite is cooled from 180°C to room temperature 20°C. A further 0.1% shrinkage occurs during cool down.

The STATE command can be entered more than once. On definition **`maxinc`** is set to the number of entered states. If **`maxinc`** is subsequently input under TERMINATION it will overwrite the value set by the STATE command. If, whilst running, all the STATE variables are processed, LUSAS will stop, independently of the value of **`maxinc`**.

14. To use the parameter exptim the total response time, ttime, and the maximum number of time steps, maxinc, must be specified under TERMINATION. The initial time step is taken as 1 day and increases according to the value of the exponential, the number of time steps (maxinc) and the total response time. The parameter exptim is only applicable to analyses using the CEB-FIP creep and shrinkage model.

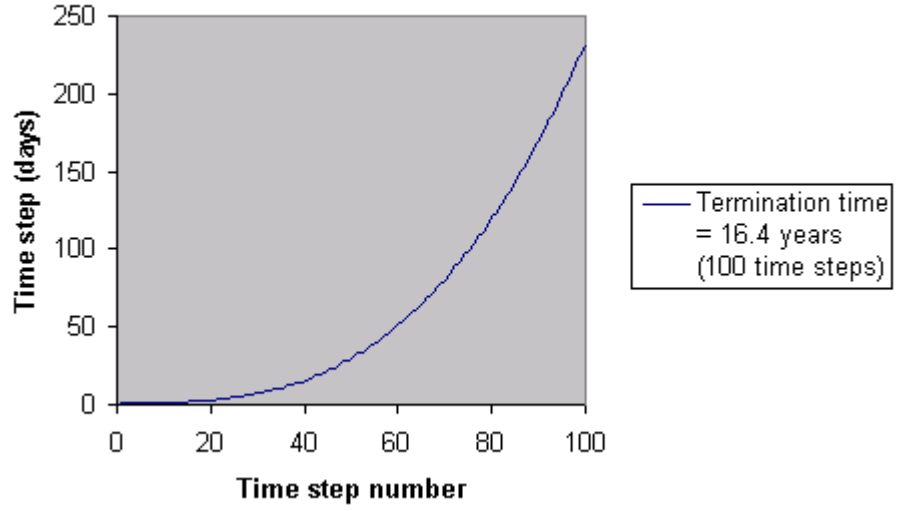


Figure 2.2 Time step number vs time step growth for an exponent of 3.0

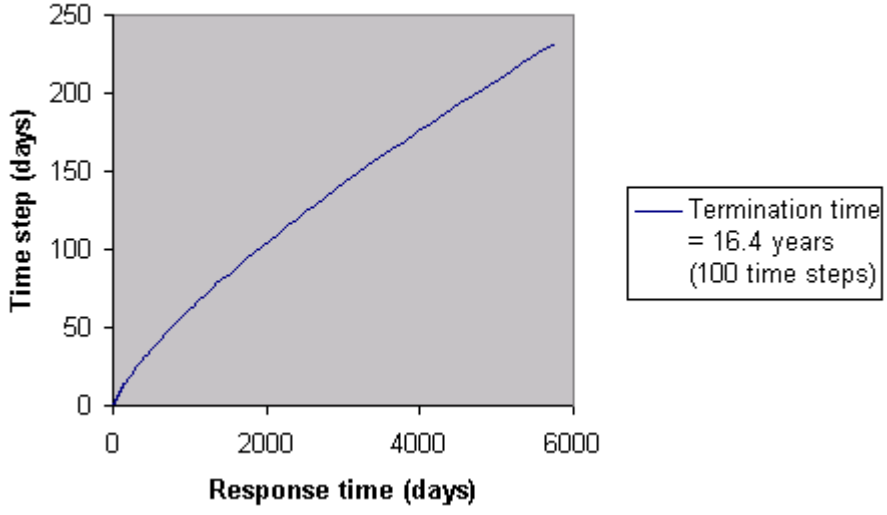


Figure 2.3 Response time vs time step growth for an exponent of 3.0

Eigenvalue Control

The EIGENVALUE CONTROL data chapter is used to control the solution procedure for eigenvalue extraction and eigenvalue buckling analyses, using either the subspace iteration method, the inverse iteration method with shifts, or the Lanczos method (standard or fast). For eigenvalue extraction using Guyan-reduced eigenvalues, see Guyan Control. The subspace iteration method is used to compute either the lowest or highest eigenmodes, while the inverse iteration method with shifts is used to compute the eigenmodes that exist within a specified eigenvalue or frequency range of interest. The Lanczos method can compute the lowest or highest eigenvalues or a specified range. For further information regarding the solution of eigenvalue problems refer to Eigenvalue Analysis, and the *LUSAS Theory Manual*.

EIGENVALUE CONTROL

either:

{CONSTANTS | SUBSPACE | LANCZOS | FAST}

or:

[INVERSE | LANCZOS | FAST] {EIGENVALUE | FREQUENCY} RANGE

[CONVERGENCE]

Eigenvalue Control

EIGENVALUE CONTROL {BUCKLING | STIFFNESS | DAMPING}

BUCKLING The BUCKLING command word is used to specify that an eigenvalue buckling problem is to be solved. This facility can only be used in conjunction with element types that support geometrically nonlinear solutions (see the *LUSAS Element Reference Manual*).

STIFFNESS The STIFFNESS command word is used to specify that an eigenvalue analysis of the stiffness matrix should be performed.

DAMPING The DAMPING command word is used to specify that the damping matrix is to be included in the analysis leading to a complex eigensolution.

With no other command words on this line (e.g. buckling/stiffness) an eigenvalue analysis using both stiffness and mass matrices will be carried out (a natural frequency analysis).

Maximal/Minimal Eigenvalues

This data section is used to specify the options for the chosen eigenvalue solution method that will solve for the lowest or highest eigenmodes in the structure. If this data section is used, the EIGENVALUE/FREQUENCY RANGE data section must be omitted. With the exception of **nroot**, default values will be assumed for all parameters that are not specified. If the CONSTANTS keyword is specified, LUSAS will select the default eigenvalue solver, which is the fast Lanczos solver unless the

analysis contains superelements, or is a branching and bracketing analysis, in which case the subspace iteration solver will be used.

```
{CONSTANTS | SUBSPACE}  nroot  [nivc  shift  norm
sturm  eigsol  maxmin  buckl]
```

or:

```
LANCZOS  nroot  [shift  norm  sturm  maxmin  buckl]
```

or:

```
FAST  nroot  [shift  norm  maxmin  buckl]
```

or for DAMPING:

```
FAST  nroot  [shiftr  shifti  norm]
```

nroot The number of eigenvalues required (must be less than or equal to the number of free nodal variables for the structure).

nivc The number of starting iteration vectors to be used (default is taken as **min** {(2***nroot**), (**nroot**+8), (number of structure free variables)}. Only used for subspace iteration.

shift The shift to be applied to the stiffness matrix (default = 0.0).

shiftr The real shift to be used in a complex (DAMPING) eigensolution (= 0.0 - not available at present).

shifti The imaginary shift to be used in a complex (DAMPING) eigensolution (= 0.0 - not available at present).

norm The normalisation procedure required for the eigenvectors (default=0).
- =0 normalisation with respect to unity
- =1 normalisation with respect to the global mass
- =2 normalisation with respect to the global stiffness

sturm Determines if a Sturm sequence check is to be applied (default=1). Automatically performed for the fast solver.
- =0 no Sturm sequence check
- =1 Sturm sequence to be carried out

eigsol Type of eigensolver required (default=0). Only used for subspace iteration.
- =0 Generalised Jacobi method
- =1 Householder-QL method

maxmin Determines whether minimum or maximum eigenvalues are required (default=0).
- =0 minimum eigenvalues required
- =1 maximum eigenvalues required

buckl Type of buckling required (default=0).
- =0 normal buckling
- =1 alternative buckling (all eigenvalues will be positive. See *Notes* on Eigenvalue Output)

Eigenvalue/Frequency Range

The EIGENVALUE RANGE or FREQUENCY RANGE data sections (for specifying eigenvalues or natural frequencies, respectively) are used to specify the options for the chosen eigenvalue solution method that will solve for a range of eigenmodes or frequencies. If this data section is used the CONSTANTS data section must be omitted. With the exception of **Rmin** and **Rmax**, default values will be assumed for all other parameters that are not specified. If neither the INVERSE, LANCZOS nor FAST keywords appear on the command line, LUSAS will select the default eigenvalue solver, which is the fast Lanczos solver unless the analysis contains superelements, or is a branching and bracketing analysis, in which case the inverse iteration solver will be used.

```
[INVERSE | LANCZOS | FAST]  {EIGENVALUE | FREQUENCY}
RANGE  Rmin  Rmax  [nreq norm]
```

Rmin The minimum eigenvalue or frequency for the defined range.

Rmax The maximum eigenvalue or frequency for the defined range (must be larger than **Rmin**).

nreq The number of eigenvalues required (default = total number of eigenvalues within the specified range).

norm The normalisation procedure required for the eigenvectors (default=0).
=0 normalisation with respect to unity
=1 normalisation with respect to the global mass
=2 normalisation with respect to the global stiffness

Convergence

The CONVERGENCE data section is used to specify the internal convergence tolerance used for the subspace and inverse iteration solvers, and controls the maximum number of iterations to be used for the standard Lanczos solver. It is not used for the fast Lanczos solver. If this data section is not specified, the default values are assumed.

```
CONVERGENCE  rtol  [nitem]
```

or:

```
CONVERGENCE  nitem
```

rtol The iterative tolerance to be satisfied (not used for either Lanczos solver)
(default for subspace iteration = 1.0E-7)
(default for inverse iteration with shifts = 1.0E-4).

For subspace iteration, this measure is defined as the absolute value of (current eigenvalue - previous eigenvalue)/current eigenvalue. This must be satisfied for all eigenvalues before convergence is assumed. For inverse iteration with shifts, this measure is defined as the tolerance for mass orthogonality.

nitem The maximum number of iterations/steps to be carried out (not used for the fast Lanczos solver)
(default for subspace iteration = 10)
(default for inverse iteration with shifts = 30)
(default for standard Lanczos = 100).

The inverse iteration procedure should always converge quickly, since the shift point is automatically updated to improve convergence if convergence difficulties are detected. The maximum number of iterations is, therefore, set to 30, since this should never be required by the algorithm. It is provided merely as a safety measure, and the default value may be changed by specifying the parameter **nitem** in the CONVERGENCE data section.

Eigenvalue Output

Eigenvalue analyses output the following results:

- Eigenvalues, frequencies and error norms for each mode requested, for example:

```
MODE   EIGENVALUE   FREQUENCY    ERROR NORM
1      81.7872      1.43934      0.2189E-11
```

- The total mass of the structure in the three global, translational directions, e.g.,

```
TOTAL MASS ACTING IN X DIRECTION = 3510.00 KG
TOTAL MASS ACTING IN Y DIRECTION = 3510.00 KG
```

- Modal displacement shapes for each mode in turn. Note that these eigenvectors have no physical meaning except to indicate the mode shape.
- Total reactions for each mode in turn (again, these have no physical meaning, since they are derived from the displacements).
- Mass participation factors are output to indicate the proportion of mass acting in each mode. Note that this is calculated automatically and there is no requirement for this factor to sum to unity. Mass participation will always be positive.

Notes

1. Combinations of the lowest, highest and a range of eigenvalues can be extracted in a single LUSAS run if required, by specifying additional EIGENVALUE CONTROL data chapters with the relevant data sections. If using the fast Lanczos solver, this must be specifed for each eigenvalue extraction, since it requires the assembly of global matrices. The standard Lanczos solver can be used during the same run as the subspace and inverse iteration solvers, if desired. If the highest eigenvalues are required, this must be specified in the last EIGENVALUE CONTROL data chapter, unless the fast Lanczos solver is in use. For example, to find the lowest two eigenvalues, the highest six eigenvalues and all frequencies in the range [0, 20] using the fast Lanczos solver, the following commands should be specified:

```
EIGENVALUE CONTROL
FAST  2
EIGENVALUE CONTROL
FAST  6  D  D  1
EIGENVALUE CONTROL
FAST FREQUENCY RANGE  0  20
```

2. Combinations of eigenvalues can also be specified for stiffness and buckling analyses by specifying additional EIGENVALUE CONTROL STIFFNESS and EIGENVALUE CONTROL BUCKLING data chapters, respectively. The same strictures apply as for the EIGENVALUE CONTROL data chapter.
3. In case the requested eigenmodes turn out not to be the ones actually needed, choosing OPTION 279 will cause the reduced stiffness and mass matrices to be saved to a restart file. They can then be accessed during a separate analysis, and more eigenmodes retrieved without needing to reduce the matrices again. Note that this facility is not available for the fast Lanczos solver.
4. The stresses from the eigenvectors may be output using ELEMENT and NODE OUTPUT CONTROL.
5. If **nmastr** master freedoms are specified within the RETAINED FREEDOMS data section, a Guyan reduction analysis will be carried out in order to obtain a first approximation to the starting iteration vectors for the subspace iteration method. Note `namast` automatically generated master freedoms will be used if **nivc** is greater than **nmast** (i.e. **namast** = **nivc** - **nmastr**).
6. If an EIGENVALUE CONTROL (including STIFFNESS and BUCKLING) data chapter using subspace iteration follows immediately after a GUYAN CONTROL data chapter, the solution from the Guyan reduction analysis will automatically be used as the first approximation to the starting iteration vectors required for the subspace iteration. This enables the Guyan solution to be improved by using the subspace iteration algorithm.
7. If an EIGENVALUE CONTROL data chapter and the GUYAN CONTROL data chapter are separated by any other data chapters, then each will be treated as a separate analysis. The same applies if a GUYAN CONTROL data chapter is immediately preceded by an EIGENVALUE CONTROL data chapter.
8. Throughout EIGENVALUE CONTROL, where a default value is shown, a **D** may be entered for a particular input variable to obtain the default value.
9. Within the subspace iteration method the eigenvalue solution may be continued after the Jacobi iteration method has failed to converge by setting Option 16, but the results should be used with caution.
10. If **nreq** is defined in the RANGE section the eigenvalues found will not always be the first eigenvalues of the system, unless the Fast Lanczos solver is used.
11. Option 230 will suppress the computation during an eigenvalue analysis. For reasonable size jobs this has the effect of reducing the plot file size to 1/3 of its original size which speeds up the analysis.
12. The Eigenvalue/Frequency Range facility cannot be used if constraint equations have been defined, unless the Fast Lanczos solver is used.

13. The Sturm sequence check may prove unreliable if constraint equations are defined in the analysis, unless the Fast Lanczos solver is used.
14. For buckling analyses involving constraint equations, the Fast Lanczos solver will only find eigenvalues either side of zero, i.e. in the range $(-\infty, 0)$ or $(0, \infty)$. If a range of eigenvalues is required in an interval which contains zero, two separate data chapters must be specified, where the interval is divided into two sub-intervals either side of zero.
15. The inverse iteration procedure should always converge quickly since the shift point is automatically updated to improve convergence if this is not the case. The maximum number of iterations is therefore set, by default, at 30 since this should never be required by the algorithm. It is provided merely as a safety measure and the default value may be changed by specifying the parameter **nitem** in the CONVERGENCE data section.

Guyan Control

The GUYAN CONTROL data chapter is used to control the solution procedure for Guyan reduced eigenvalue extraction analyses (for eigenvalue analyses using subspace iteration see Eigenvalue Control). For further information regarding the solution of eigenvalue problems by Guyan reduction refer to Guyan Reduction, and the *LUSAS Theory Manual*.

```
GUYAN CONTROL  [BUCKLING | STIFFNESS]
```

BUCKLING The BUCKLING command word is used to specify that the eigenvalue buckling problem is to be solved using Guyan reduction. This facility must only be used in conjunction with element types that support nonlinear solutions (see the *LUSAS Element Reference Manual*).

STIFFNESS The STIFFNESS command word is used to specify that an eigenvalue analysis of the stiffness matrix is to be carried out using Guyan reduction.

Constants

The CONSTANTS data section is used to specify the constants for the Guyan reduction as well as to define the options available to the user. This section is compulsory, and with the exception of **nroot** default values will be assumed for all parameters that are not specified.

```
CONSTANTS  nroot  [namast shift norm sturm eigsol
maxmin buckl]
```

nroot The number of eigenvalues required (must be less than or equal to the number of free nodal variables of the reduced structure).

namast The number of automatically generated master freedoms (default=0) (see Retained Freedoms).

shift The shift to be applied to the eigenvalue procedure (default = 0.0).

norm The normalisation procedure required for the eigenvectors (default=0).
=0 normalisation with respect to unity
=1 normalisation with respect to the global mass
=2 normalisation with respect to the global stiffness

sturm Determines if a Sturm sequence check is to be applied (default=1).
=0 no Sturm sequence check
=1 Sturm sequence to be carried out

eigsol Type of eigensolver required (default=0).
=0 Generalised Jacobi method
=1 Householder-QL method

maxmin Determines whether minimum or maximum eigenvalues are required (default=0).
=0 minimum eigenvalues required
=1 maximum eigenvalues required

buckl Type of buckling required (default=0).
=0 normal buckling
=1 alternative buckling (all eigenvalues will be positive) (see *Notes*).

Notes

1. Specified master freedoms may also be input via the RETAINED FREEDOM data chapter.
2. Master freedom selection may be either totally specified, or totally automatic, or a combination of the two. LUSAS will confirm the type of master freedoms used in the operation, and in the case of wholly specified or wholly automatic, will warn to that effect.
3. If LUSAS is unable to create the full requested number of masters a warning to that effect will be invoked, and the solution will proceed with the reduced number of automatic master freedoms. Note that this is equivalent to carrying out an eigen-analysis of the full problem.
4. The solution obtained by a Guyan reduction can be improved upon by employing the subspace iteration algorithm. This will be done if an EIGENVALUE CONTROL data chapter follows immediately after a GUYAN CONTROL data chapter; the Guyan reduction solution will automatically be used as the first approximation to the starting iteration vectors required for the subspace iteration.
5. If an EIGENVALUE CONTROL data chapter and the GUYAN CONTROL data chapter are separated by any other data chapter then each will be treated as a separate analysis. The same applies if a GUYAN CONTROL data chapter is immediately preceded by an EIGENVALUE CONTROL data chapter.
6. Throughout GUYAN CONTROL, where stated, a D may be entered for a particular input variable to obtain the default value.
7. The nonlinear solution may be continued after the Jacobi iteration method has failed to converge by setting Option 16.
8. Freedoms featuring in constraint equations will not be considered for use as automatic masters.

9. The Sturm sequence check may prove unreliable if constraint equations are defined in the analysis.

Modal Damping Control

The MODAL_DAMPING CONTROL data chapter is used to control the computation of viscous or structural distributed damping factors where element contributions towards damping vary within the structure. This data chapter should always follow an eigenvalue analysis since this analysis depends upon the eigensolutions that should have already been obtained. This section is valid for viscous (modal) and structural (hysteretic) damping. After this section LUSAS computes the distributed viscous and/or structural damping factors and echoes the damping factors to the LUSAS output file. These damping factors may subsequently be used in a harmonic or spectral response analysis or a LUSAS Modeller modal analysis session.

```
MODAL_DAMPING CONTROL  [VISCOUS | STRUCTURAL]
```

Constants

The CONSTANTS data section is used to specify the default damping factor for a mode if the distributed damping factor is not to be computed (i.e. mode omitted from MODES section). If this section is not specified the system default values will apply.

```
CONSTANTS  damp
```

damp The overriding default damping factor if the distributed damping factor is not to be computed (default=0.05 for VISCOUS damping, default=0.00 for STRUCTURAL damping)

Material Properties

This line is optional and is only required if the damping parameters from the MATERIAL PROPERTY data chapter and not the DAMPING PROPERTIES data chapter are to be used for the computation of the distributed damping factors. By default Rayleigh parameters from the DAMPING PROPERTIES section are used. If no DAMPING PROPERTIES are input then Rayleigh parameters from the MATERIAL PROPERTIES section are used.

```
MATERIAL PROPERTIES
```

Modes

This section is optional and is used when the distributed damping is to be computed for only some modes. By default distributed damping factors will be computed for all modes.

```
MODES
modei
```

modei Number of the (i)th mode for which distributed damping must be computed. This line must be repeated for each mode but may be generated by using the LUSAS data generation structure FIRST, INC, INC etc.

Distributed damping factors will not be computed for modes omitted here but these modes will take the default damping factor as specified in the CONSTANTS data section.

Notes

1. The parameter norm in EIGENVALUE or GUYAN control must be set to 1 in the eigenvalue analysis to normalise the eigenvectors with respect to global mass before computation of distributed modal damping factors.
2. After computing the distributed viscous and/or structural damping factors LUSAS will write the values to the output file.

Example. Damping Properties

```
    :
C   Table of viscous damping properties
DAMPING PROPERTIES VISCOUS
C   a_i,    b_i,    ω_i,       a_i,    b_i,    ω_i,       a_i,
b_i,    ω_I  (Hz)
 1 3 1    0.1    0.05   0        0.1    0.05   0.2
0.3    0.2    0.3 ...
    0.1    0.05   0.5      0.1    0.05   0.6       0.3
0.2    0.7 ...
    0.3    0.2    0.9      0.1    0.05   1.0       0.1
0.05   1.5
C   Table of structural damping properties
DAMPING PROPERTIES STRUCTURAL
C   a_i,    b_i,    ω_i,       a_i,    b_i,    ω_i,       a_i,
b_i,    ω_I  (Hz)
 1 0 0    0.1   0.05   0.1       0.1   0.05    0.2
0.3    0.4    0.4 ...
    0.1   0.05   0.5       0.1   0.05    0.6       0.5
0.2    0.7 ...
    0.7   0.2    0.9       0.1   0.05    1.0       0.1
0.05   1.5
```

```
 2 0 0    0.1   0.05   0.1       0.1   0.05    0.2
0.3    0.4    0.4 ...

    0.1   0.05   0.5       0.1   0.05    0.6       0.5
0.2    0.7 ...

    0.7   0.2    0.9       0.1   0.05    1.0       0.1
0.05   1.5

 3 0 0    0.1   0.05   0.1       0.1   0.05    0.2
0.3    0.4    0.4 ...

    0.1   0.05   0.5       0.1   0.05    0.6       0.5
0.2    0.7 ...

    0.7   0.2    0.9       0.1   0.05    1.0       0.1
0.05   1.5

    :

LOAD CASE

 CL

  4 16 4  5./32.**.5

C   Eigenvalue analysis (3 modes)

EIGENVALUE CONTROL

 CONST    3  3  D  1

C   Compute distributed viscous damping factors for all
three modes. Rayleigh

C   parameters from the DAMPING PROPERTIES VISCOUS table
will be used in the

C   computation.

MODAL_DAMPING CONTROL VISCOUS

C   Compute distributed structural damping factors for
modes 1 and 3. Use the

C   overriding default damping constant of 0.33 for mode
2. Rayleigh parameters from

C   the DAMPING PROPERTIES STRUCTURAL table will be used
in the computation.

MODAL_DAMPING CONTROL STRUCTURAL

 CONSTANTS   0.33

 MODES

    1 3 2
```

```
C   Harmonic response analysis using distributed viscous
damping factors and system
C   factors (no distributed structural damping factors
will be used).
C   Default values for structural damping are used.
HARMONIC_RESPONSE CONTROL
 CONSTANTS   3   0.1
 VISCOUS DAMPING DISTRIBUTED
 FREQUENCIES
    0.01
    0.175
    0.477
 HARMONIC LOADING REAL 1
 HARMONIC LOADING IMAGIN 1
END
```

Spectral Control

The SPECTRAL CONTROL data chapter is used to control the solution procedure for spectral response analyses. This data chapter must always follow an eigenvalue analysis since the analysis depends upon the eigen-solutions that should have already been obtained. For further information regarding the solution of spectral response problems refer to Spectral Response Analysis, and the *LUSAS Theory Manual*.

```
SPECTRAL CONTROL
```

Constants

The CONSTANTS data section is used to specify certain constants for the spectral response procedure. This data section is mandatory.

```
CONSTANTS  nmod  icmb  [dampm]
```

nmod The number of modes to be processed. Must be less than or equal to the number of eigenvalues previously extracted (default = number of eigenvalues previously solved for, i.e. **nroot**).

icmb The spectral combination type.

=0 for no combination required (default)
=1 for SRSS (square root of the sum of the squares)
=2 for CQC (complete quadratic combination)
=3 for Absolute Sum

dampm The overriding default value for the modal damping coefficients not specified in the MODAL DAMPING data section. (default = 0.05, that is 5%).

Modal damping

The MODAL DAMPING data section is used to specify the damping of each system mode of vibration as a portion of the critical damping for that mode. Modal damping may be input for each mode directly (LUSAS will then assign the modal damping value to the modal frequency when it has been determined by the eigenvalue analysis) using MODES, or may be input at known frequencies (LUSAS will then interpolate to the modal frequencies computed in the eigenvalue analysis) using FREQUENCIES. Alternatively the physical distribution of damping in a structure may be modelled using MODAL DAMPING DISTRIBUTED, where the computed modal damping factors from MODAL_DAMPING CONTROL VISCOUS (see Modal Damping Control) is used.

The following apply only for MODES and FREQUENCIES keywords:

```
MODAL DAMPING  {DISTRIBUTED | MODES  RAYLEIGH |
FREQUENCIES RAYLEIGH}

{Mi | ωi}  ai  [bi]
```

Mi The (i)th mode number.

ωi The (i)th specified frequency.

ai The modal damping value for the (i)th mode or (i)th specified frequency, or the mass Rayleigh damping constant when the RAYLEIGH option is invoked.

bi The stiffness Rayleigh damping constant when the RAYLEIGH option is invoked.

Excitation

The EXCITATION data section specifies the excitation direction and is compulsory.

```
EXCITATION  Xdir  Ydir  Zdir
```

Xdir The excitation factor in the global X-direction

Ydir The excitation factor in the global Y-direction

Zdir The excitation factor in the global Z-direction

Note. The above factors are the components of the vector defining the direction of excitation.

Spectral Curve

The SPECTRAL CURVE data section is used to specify the frequency-displacement, frequency-velocity or frequency-acceleration curve.

SPECTRAL CURVE nspts icurve

< $freq_i$ $value_i$ >$_{i=1,n}$

nspts The number of points defining the spectral curve.

Icurve The spectral curve type:
- =1 for frequency-displacement curve
- =2 for frequency-velocity curve
- =3 for frequency-acceleration curve
- =4 for period displacement curve
- =5 for period velocity curve
- =6 for period acceleration curve

$freq_i$ The frequency for the (i)th point on the spectral curve.

$value_i$ The value of the displacement/velocity/acceleration for the (i)th point on the spectral curve in ascending order.

Notes

1. The parameter `norm` in EIGENVALUE or GUYAN CONTROL must be set to 1 in the eigenvalue analysis to normalise the eigenvectors to global mass before a spectral analysis can be carried out.
2. To use distributed damping factors MODAL DAMPING DISTRIBUTED must be specified, otherwise the current direct input damping factors will be used.
3. When MODAL DAMPING is specified it must follow CONSTANTS.
4. If PLOT FILE is placed after the SPECTRAL CONTROL chapter the spectral results are transferred to the LUSAS Modeller plot file for subsequent plotting.
5. Damping is only taken into account for spectral combination type CQC.
6. Spectral response may also be carried out in LUSAS Modeller.

Harmonic Response Control

The HARMONIC_RESPONSE CONTROL data chapter is used to control the solution procedure for forced vibration or harmonic response analysis. This data chapter should always follow an eigenvalue analysis since this analysis depends upon the eigensolutions that should have already been obtained. For further information regarding the solution of forced vibration problems refer to Harmonic Response Analysis, and the *LUSAS Theory Manual*.

HARMONIC_RESPONSE CONTROL

Constants

The CONSTANTS data section is used to specify the harmonic response control parameters and the user defined default damping values. This section may only be specified once in an analysis; once these constants have been defined they are unchangeable.

```
CONSTANTS  nmod dampm damps
```

nmod The number of system eigenmodes which are to be utilised in the harmonic response analysis. (default = number of eigenvalues previously solved for, i.e. **nroot**)

dampm The overriding default value for the viscous damping coefficients not specified in the VISCOUS DAMPING data section. (default = 0.05, that is 5%)

damps The overriding default value for structural damping coefficients not specified in the STRUCTURAL DAMPING data section. (default = 0.0, that is 0%)

Frequencies

The FREQUENCY data section is used to specify the loading (or sampling) frequencies for which the harmonic analysis will be carried out.

```
FREQUENCIES
freqi
```

freqi Value of the (i)th sampling frequency. This line must be repeated for each frequency but may be generated by using the LUSAS data generation structure FIRST, INC, INC.

Viscous Damping

The VISCOUS DAMPING data section is used to specify the damping of each system mode of vibration as a portion of the critical damping for that mode. Viscous damping may be input for each mode directly using the MODES keyword (LUSAS will then assign the viscous damping value to the modal frequency when it has been determined by the eigenvalue analysis) or may be input at known frequencies (LUSAS will then interpolate to the modal frequencies computed in the eigenvalue analysis) using the FREQUENCIES keyword. Alternatively the physical distribution of damping in a structure may be modelled using VISCOUS DAMPING DISTRIBUTED, where the computed viscous damping factors from MODAL_DAMPING CONTROL VISCOUS are used (see section on Modal Damping Control).

The following apply only for MODES and FREQUENCIES keywords:

```
VISCOUS DAMPING  {DISTRIBUTED | MODES  RAYLEIGH |
FREQUENCIES  RAYLEIGH}

{Mi | ωi}  ai  [bi]
```

Mi The (i)th mode number.

ωi The (i)th specified frequency.

ai The modal damping value for the (i)th mode or (i)th specified frequency, or the mass Rayleigh damping constant when the RAYLEIGH option is invoked.

bi The stiffness Rayleigh damping constant when the RAYLEIGH option is invoked.

Structural Damping

The STRUCTURAL DAMPING data section specifies the damping of each system mode of vibration via the hysteretic damping value; this value may be different for each system mode but is constant for the structure for each mode. Structural damping may be input for each mode directly (LUSAS will then assign the structural damping value to the modal frequency when it has been determined by the eigenvalue analysis) using the MODES keyword or may be input at known frequencies using the FREQUENCIES keyword (LUSAS will then interpolate to the modal frequencies computed in the eigenvalue analysis). Alternatively the physical distribution of damping in a structure may be modelled using STRUCTURAL DAMPING DISTRIBUTED, where the computed modal damping factors from MODAL_DAMPING CONTROL STRUCTURAL are used (see section on Modal Damping Control).

The following apply only for MODES and FREQUENCIES keywords:

STRUCTURAL DAMPING {DISTRIBUTED | MODES | FREQUENCIES}

{Mi | ωi} μi

Mi The (i)th mode number.

ωi The (i)th specified frequency.

μi The structural damping value for the (i)th mode or (i)th specified frequency.

Harmonic Loading

The HARMONIC LOADING data section specifies the harmonic loads in terms of their **real** and **imaginary** components. By specifying both components the phase difference of the loads can be specified.

HARMONIC LOADING {REAL | IMAGINARY} lnum

lnum The LOAD CASE number which describes the harmonic loading component.

The HARMONIC LOADING defaults are defined as follows:

- If no HARMONIC LOADING is defined or if the HARMONIC LOADING command is specified without any parameters then the default loading is assumed: LOAD CASE number 1 is used as the real component and the imaginary component is assumed to be zero;
- If only the real component of load is specified then the imaginary component is assumed to be zero;
- If only the imaginary component of load is specified then the real component is assumed to be zero.

Notes

1. The parameter **norm** in EIGENVALUE or GUYAN CONTROL must be set to 1 in the eigenvalue analysis to normalise the eigenvectors to global mass before a harmonic response analysis can be carried out.
2. Modal and structural damping values for unspecified system modes for direct input of damping using VISCOUS DAMPING MODES or FREQUENCIES will be interpreted as the default values in the CONSTANTS data section.
3. Modal and structural damping values specified at known FREQUENCIES will be interpolated at the **nmod** selected system frequencies: interpolation to system frequencies within the range specified in the relevant DAMPING data section will be linear, while interpolation to system frequencies beyond the range specified will be constant and will invoke a warning message.
4. Original and interpolated damping will be echoed.
5. When Rayleigh damping constants are specified the corresponding modal damping is computed from:

$$\text{modal damping} = \frac{\left(\frac{a_r}{w} + \frac{w}{b_r}\right)}{2}$$

   where w is either the natural frequency corresponding to the particular mode or the frequency that is input on the data line. This conversion is done before the interpretation of the damping is carried out.
6. When the RAYLEIGH option is used notes 3. and 4. will still apply.
7. To use distributed damping factors VISCOUS and/or STRUCTURAL DAMPING DISTRIBUTED must be specified otherwise the current direct input damping factors will be used.
8. LOAD CASE numbering is assumed to be consecutive. After a RESTART command has been used the new LOAD CASES start renumbering from 1 again.
9. The harmonic loading must be respecified with every new HARMONIC_RESPONSE data command.
10. Throughout HARMONIC_RESPONSE CONTROL, where a default is given, you can input a **D** for a particular input variable to invoke the default value.
11. Option 181 provides output in the polar coordinate system.
12. PDSP loading is not permitted with this facility.

Fourier Control

The FOURIER CONTROL data chapter controls the input of the Fourier components for use with Fourier elements. This data chapter must be used if Fourier elements are

utilised. For further information regarding the solution of Fourier problems refer to the section titled Fourier Analysis, and the *LUSAS Theory Manual*.

```
FOURIER CONTROL
HARMONIC  [symmetry]
H Hlast Hdiff
```

symmetry The harmonic components to be analysed (default = 0):
= 0 both symmetric and asymmetric components
= 1 only the asymmetric components
= 2 only the symmetric components

H Hlast Hdiff The first harmonic, last harmonic and difference in harmonics of the Fourier expansion.

Example. Fourier Control

In order to apply a line load with an intensity of 8 per unit length of the structure over an arc from 10° to 20° and to solve for the first 4 harmonics (symmetric), the datafile will have the following form:

```
LOAD CASE
    CL
    1  0  0   16.   0.   0.
CURVE DEFINITION  4  USER   :define User curve 4
    10  1       :assign a value of 1 at 10°
    20  1       :assign a value of 1 at 20°
CURVE ASSIGNMENT
    1  0.5  4         :assign load curve 4 with a factor
of 0.5 to loadcase 1
FOURIER CONTROL
    HARMONICS  2      :solve symmetric components only
    0  3  1           :evaluate harmonics 0,1,2 and 3.
```

Concentrated loads, constant body forces and body force potentials are applied in the global XYZ directions, as opposed to surface tractions, initial stresses, initial strains and thermal loading which are all applied in the local xyz directions. Note that concentrated loads/nodal reactions are input and output as forces per unit length. Concentrated loads may also be applied in the cylindrical coordinate system by setting Option 202.

Notes

1. Both the symmetric and the asymmetric components will be calculated unless overridden by the **symmetry** input parameter.
2. The HARMONIC data line applies to all **H**, **Hlast**, **Hdiff** series following the specification of this command. HARMONIC may be respecified in this data chapter if required.
3. All loads for a static analysis will be decomposed into the defined harmonic components.
4. If a dynamic, eigenvalue or harmonic response analysis is required the harmonic series should only contain 1 term.
5. Concentrated loads may be applied in the local cylindrical coordinate system by using Option 202.

Creating a Plot File

The PLOT FILE data section writes the analysis mesh data and results to secondary storage for subsequent plotting using LUSAS Modeller. For nonlinear, dynamic, transient and viscous analyses the frequency of plot file generation is controlled via the OUTPUT data sections (in these cases the PLOT FILE data section is not required).

```
PLOT FILE
```

Notes

1. For linear analyses PLOT FILE must be placed after all the LOAD CASE and ELEMENT/NODAL OUTPUT commands.
2. PLOT FILE should be specified prior to ENVELOPE and LOAD COMBINATION.
3. To save eigenvalue results, place the PLOT FILE command prior to the harmonic or spectral commands.
4. The structural definition can be saved (and subsequently viewed) when using a pre-analysis data check (Option 51).
5. When carrying out a nonlinear, dynamic, transient or a viscous analysis the plot file should be requested by setting the **incplt** parameter within the OUTPUT data section in the appropriate data chapter. However, the PLOT FILE command may still be specified at the end of the analysis.
6. If a nonlinear, dynamic, transient or a viscous analysis terminates and plot files have been requested a plot file will always be created for the last increment.

Restart Facilities

Data can be written to disk to enable a nonlinear, dynamic, transient or viscous problem to be restarted from additional steps or a load combination to be carried out.

Note that for nonlinear, dynamic, transient or viscous analyses the frequency of restart file generation is controlled via the OUTPUT data sections of the appropriate CONTROL data chapter.

Writing to a Restart File

The data section RESTART WRITE is used to control writing of the LUSAS database to disk for subsequent processing.

```
RESTART WRITE  [BRIEF]  [ndump]
```

ndump The number of the restart dump to be written.

Notes

1. Restart files are written to a file of the same name as the data file but with an extension of **.rst**.
2. May be placed after any LOAD CASE data chapter for nonlinear problems.
3. The RESTART WRITE BRIEF command will dump sufficient information to enable a problem to be restarted for LOAD COMBINATIONS. This option cannot be used for restarting any other problem.
4. If no dump number is specified the dump number will be incremented by one from the previous highest value.
5. If a dump number is specified that already exists, a warning message will be printed and the dump will be overwritten.
6. In a nonlinear problem a dump will only be written if the solution converged.
7. When carrying out a nonlinear, dynamic, transient or a viscous analysis the restart write file should be requested by setting the **incrst** parameter within the OUTPUT data section in the appropriate data chapter.
8. If a nonlinear, dynamic, transient or a viscous analysis terminates and restart dumps have been requested a restart dump will always be created for the last increment.
9. Selecting Option 251 will generate a restart file from a data check phase. To complete the analysis after a successful data check, initiate the restart file and LUSAS will recommence the analysis at the equation solution phase. This particular restart facility is only valid for a linear static problem and is not available for other types of analyses.

Reading from a Restart File

The data section RESTART READ is used to control reading of the written LUSAS database from disk for subsequent processing.

```
RESTART READ  [ndump]  [fname]
```

ndump The number of the RESTART WRITE dump to be recovered (default = last saved dump).

fname The file name of the restart file (without extension). (default = name of data file).

Notes

1. The filename extension **.rst** is used for all restart files.
2. The RESTART READ data command must be the first line of any restart processing (unless a SYSTEM command is present in which case the RESTART READ will follow this).
3. Further data may be stored in the restart file after a RESTART READ.
4. A new LUSAS Modeller plot file will be created in a nonlinear, dynamic, transient or viscous analysis if required after a RESTART READ. The existing plot file can be used by setting Option 204 either in the previous analysis or in the restart data file.
5. Any system parameters specified in the previous analysis must be respecified on restart.
6. A RESTART analysis can be utilised to create a LUSAS Modeller plot file only.

Example. Restart Read

To create a plot file only:

```
RESTART READ
PLOT FILE
END
```

To continue a nonlinear analysis:

```
RESTART READ
LOADCASE
```

```
...
```

```
NONLINEAR CONTROL
```

```
...
```

```
END
```

To perform a pre-solution restart analysis created using Option 251:

```
RESTART READ filename
PLOT FILE
END
```

Re-Solution

The reduced stiffness matrices for linear elastic analyses can be saved to disk by specifying option 279 when using the restart facility. This allows subsequent analyses to process additional load cases using the matrix stored on disk to reduce solution times. The stiffness file takes a default .stf extension, and the location of the stored stiffness file can be changed using the STIFFNESS environment variable. The stiffness file changes during the re-solution phase. The highest load case from the previous re-solution is stored in the file. Loadcases will be numbered in ascending order if not otherwise specified on the LOAD CASE line, as is the default when the data file is tabulated by Modeller. The stiffness file must have the same name as the restart file, but with a .stf extension.

The stiffness matrix is accessed for re-solution using the RESTART READ command, and it must be available when the restart file is read if option 279 was used when the restart file was created. If LUSAS cannot locate a stiffness file while reading the restart file, an error will be issued. If the stiffness file has accidentally been erased or corrupted, it should be recreated by re-running the original analysis.

The data syntax for creating a stiffness file using the restart write command is as follows:

```
problem title re-solution
options  279
...
load case
...
restart write
end
```

The stiffness file is read using the restart read command and subsequent load cases are processed by specifying the following syntax:

```
restart read filename
load case
...
load case
...
load case
...
end
```

Notes

1. It is advisable to always work on a backup copy of the stiffness file as it can become corrupted if LUSAS terminates with an error while accessing the file.
2. The stiffness file cannot be split using the method outlined for splitting the frontal file.

End

The data section END is used to terminate the current problem data file and is mandatory.

```
END
```

Notes

1. The END statement must be the last data section in the analysis datafile.

Appendix A LUSAS User Options

For usability LUSAS User Options are listed by option number and also by Category

LUSAS User Options

No.	Effect of option (plus related notes where applicable)	Category
2	Suppress node coordinate checks for similar coordinates and missing nodes	INPUT
4	No overwriting present in data exit on first occurrence	INPUT
14	Compute element stresses and strains (plates/flat shells)	PROCESSING
16	Continue nonlinear/eigenvalue solution when failed to converge	NONLINEAR / EIGENVALUES
17	Keep global matrix assembly in-core for fast/iterative solvers	NONLINEAR
18	Invokes finer numerical integration rules for elements	INTEGRATION
19	Invokes coarse numerical integration rule for semiloof	INTEGRATION
20	Invokes non-orthogonal concrete crack model (concrete model 24)	INTEGRATION
22	Invokes fracture energy strain-softening model (concrete model 24)	INTEGRATION
23	Invokes strain-variable shear retention model (concrete model 24)	INTEGRATION
26	Reduce number of lines output in coordinates	OUTPUT
30	Output final node coordinates in ascending order	OPTIMISATION
32	Suppresses stress output but not stress resultant output	OUTPUT
33	Output direction cosines of local cartesian systems for interface	OUTPUT
34	Output element stress resultants	OUTPUT
36	Follower loads (available for selected elements)	NONLINEAR
39	Stress smoothing for rubber material models	OUTPUT
40	Output nodal displacement increments for nonlinear analyses	OUTPUT
42	Output nodal residual forces (nonlinear)	OUTPUT

No.	Effect of option (plus related notes where applicable)	Category
44	Suppress expanded input data printout except load cases	OUTPUT
45	Suppress expanded input data printout for load cases	INPUT
46	Suppress page skip between output stages	OUTPUT
47	Axisymmetry about the global X-axis	INPUT
48	Switch CBF input from Force/unit volume to acceleration	INPUT
49	Automatic correction of midside nodes	INPUT
51	Data processing only	INPUT
53	Positive definite/singularity check on modulus matrix at each Gauss point	PROCESSING
54	Updated Lagrangian geometric nonlinearity	NONLINEAR
55	Output strains as well as stresses	OUTPUT
59	Output local direction cosines for shells	OUTPUT
61	All 2D slideline surfaces defined with outward normals	INPUT
62	Continue solution if more than one negative pivot occurs	NONLINEAR
64	Non-symmetric frontal solution	SOLVER
66	Calculate element internal forces	PROCESSING
70	Echo nodal data read from data transfer file	COUPLED
72	Suppress machine code inner loops	OPTIMISATION
77	Output principal stresses and directions for solids	OUTPUT
87	Total Lagrangian geometric nonlinearity	NONLINEAR
91	Formulate element mass with fine integration	INTEGRATION
93	Suppress intermediate eigenvalue output for Subspace or bracketing methods	EIGENVALUES
100	Output optimum frontal solution order	OPTIMISATION
102	Switch off load correction stiffness matrix due to centripetal acceleration	LOADING
105	Lumped mass matrix	DYNAMICS
110	Use assumed shear strain field for QTS8 thick shell elements	PROCESSING
111	Don't include superelement generation load	PROCESSING
115	Output displacements and reactions in transformed axes	OUTPUT
116	Suppress the output of internal constraint forces	OUTPUT
117	Invokes geometric assignments input	INPUT
118	Invokes material assignments input	INPUT
119	Invokes temperature input for joints (by default no temperature in joints)	INPUT
123	Clockwise node numbering	INPUT
124	Save shape function array for restarts	RESTART
131	Parallel frontal solver algorithm	SOLVER
134	Gauss to Newton-Cotes integration for elements	INTEGRATION
136	Stress resultants by f=Kd and cubic interpolation	PROCESSING
137	Stress resultants by f=Kd and equilibrium	PROCESSING
138	Output yield flags only	NONLINEAR

No.	Effect of option (plus related notes where applicable)	Category
139	Output yielded, cracked or crushed integration points only for MNL	NONLINEAR
143	Output shear forces in plate bending elements.	OUTPUT
144	Output element results for each load case separately	OUTPUT
146	Include stiffness (second) Rayleigh damping parameter	DYNAMICS
147	Omit output phase	OUTPUT
155	14 point integration rule for HX20	INTEGRATION
156	13 point integration rule for HX20	INTEGRATION
157	Material Model 29 required (mandatory with non-cross-section elements)	NONLINEAR
164	Guide arc-length solution with current stiffness parameter	NONLINEAR
167	Eulerian geometric nonlinearity	NONLINEAR
169	Suppress extrapolation of stress to nodes for semiloof shell	OUTPUT
170	Suppress storage of shapes	OPTIMISATION
171	Use standard shear strain field for QTS4 thick shell elements	INTEGRATION
172	Formulate modulus matrix by integrating across the cross-section	INTEGRATION
179	Verify arguments to user-supplied constitutive model routines	NONLINEAR
181	Harmonic response/complex mode output required in polar system	OUTPUT
183	Double convergence check on coupled analyses running concurrently	COUPLED
184	Suppress stringent slave search for slideline problems	SLIDELINES
185	Suppress initial slideline_surface stiffness check	SLIDELINES
186	Suppress initial penetration check for slidelines	SLIDELINES
187	Include all variables in convergence norm computations	NONLINEAR
197	Fast element stiffness formulation without sparsity checks	OPTIMISATION
202	Apply CLs in local cylindrical coordinates with Fourier elements	LOADING
204	Use existing Plot file after restarting an analysis	RESTART
207	Use angle in degrees to define direction of anisotropy for user-defined materials	INPUT
208	Write plot file in pre version 11.0 format (not valid for explicit dynamics)	OUTPUT
225	Use alternative number of parameters for enhanced strain interpolation function	INTEGRATION
227	Error if a value is outside the bounds of a table of values	PROCESSING
229	Co-rotational nonlinear geometric formulation	NONLINEAR
230	Suppress stress calculation for eigenvalue analysis	OPTIMISATION
231	Error for material direction cosines not in plane of semiloof elements	PROCESSING
235	Heat fraction specified for thermal softening	INPUT

No.	Effect of option (plus related notes where applicable)	Category
242	Temperatures input and output in degrees Celsius	INPUT
247	Extra output for iterative solvers	SOLVER
248	Single precision preconditioning for PCG solver	SOLVER
250	Datafile has been created by Modeller	INPUT
251	Data processing and create a restart file (linear static analyses only)	INPUT
252	Suppress pivot warning messages from frontal solution algorithm	SOLVER
253	Emissivity specified in thermal environment properties	INPUT
254	Output all eigensolutions computed using inverse iteration	EIGENVALUE
255	Output all radiation surface view factors to FACET file.	OUTPUT
256	Suppress recalculation of radiation view factors in coupled analyses	COUPLED
259	Save nodal stresses in plot file to avoid stress extrapolation in Modeller	OUTPUT
261	Switch for selecting root with lowest residual norm (arc-length method)	NONLINEAR
266	Layer by layer computation of mass matrix for solid composites	INTEGRATION
272	Activate or deactivate elements	PROCESSING
273	Contact penalty stiffness update for augmented Lagrangian and slidelines	NONLINEAR
274	Use Uzawa rather than Powell method for node activity	PROCESSING
275	Automatic time step calculation for implicit dynamics	DYNAMICS
278	Assign six degrees of freedom to all thick shell element nodes	PROCESSING
279	Save the reduced stiffness for further use in a linear analysis	SOLVER
281	Autoloader file	PROCESSING
289	Plastic work is to be transferred in a coupled analysis	COUPLED
290	Write element stiffness and mass matrices to plot file	OUTPUT
303	Exclude incompatible modes for solid composite elements	PROCESSING
304	Plot/restart databases open until end of analysis. Speeds subsequent dumps.	OUTPUT
308	Arc-length solution using local relative displacements (interface elements)	NONLINEAR
310	Output results to user-defined output file	OUTPUT
311	Compute number of eigenvalues in specified range and terminate analysis	EIGENVALUE
318	Adaptive Analysis: Overwrite disp's with interpolated disp's from old mesh	PROCESSING
319	Invokes eccentricity input for QTS4 thick shell family	INPUT
320	Global matrix (and righthand side) assembly and output only	OUTPUT

No.	Effect of option (plus related notes where applicable)	Category
323	Extra preconditioning for iterative solver with a hierarchical basis	SOLVER
324	Specify damping properties for joint properties general explicitly	INPUT
340	Allow application of angular velocities with nonlinear auto incrementation	NONLINEAR
342	Always output progress information during solution	OUTPUT
344	Suppress condition number estimate in fast solvers	SOLVER
345	Separate plot file for each analysis step	OUTPUT
346	Single segment contact from beginning of problem	PROCESSING
347	Single segment contact from beginning of second increment	PROCESSING
348	Slideline summary output	OUTPUT
350	Invokes product moment of inertia (Iyz) input for BTS3 nonlinear thick beam elements	INPUT
352	Turn off consistently linearised formulation for slidelines	SLIDELINES
353	Use of a fixed penalty parameter with slidelines	SLIDELINES
355	Slideline Cohesion with Coulomb friction in 2D	SLIDELINES
363	Suppress echo of computed laminate properties in output file (woven fabric material)	OUTPUT
364	Write element stiffness, mass and damping matrices to plot file	OUTPUT
370	Strain hardening approach to modified von Mises and Hoffman material models (77/78)	NONLINEAR
373	Extended input data diagostics for CEB-FIP creep and shrinkage model	OUTPUT
376	Suppress use of METIS ordering for fast solvers	SOLVER
377	Use minimum amount of memory for BCS solvers	SOLVER
380	Output stress resultants relative to beam axes for eccentric BTS3 elements	OUTPUT
381	Apply CBF,ELDS,UDL loads along beam axes with BMS3 elements	LOADING
385	Preserve loading assigned to deactivated elements for subsequent application on activation	LOADING
386	Use of weighted force/stiffness distribution with slidelines	SLIDELINES
387	Redefine NSET to angle in element xy-plane for use in orthotropic material properties	PROCESSING
388	Use duplicate nodes facility for collapsed elements	PROCESSING
389	Echo thermal link elements to output file	OUTPUT
390	Include selected results in plot file for data-only-processing	OUTPUT
394	Lamina directions supported	PROCESSING
395	Use 14-point fine integration rule for mass matrix of TH10 family (used together with 91)	INTEGRATION

No.	Effect of option (plus related notes where applicable)	Category
396	Improved top/middle/bottom transverse shear stress calculation for thick shell elements	INTEGRATION
397	With data-only-processing, compute memory requirements for solution of equations	PROCESSING
398	Use all 27/18 integration points for stress extrapolation for 20/16 node solids with fine integration	INTEGRATION
399	Retain inactive elements in birth and death solution (old analysis type)	PROCESSING

LUSAS User Options by Category

No.	Effect of option (plus related notes where applicable)	Category
70	Echo nodal data read from data transfer file	COUPLED
183	Double convergence check on coupled analyses running concurrently	COUPLED
256	Suppress recalculation of radiation view factors in coupled analyses	COUPLED
289	Plastic work is to be transferred in a coupled analysis	COUPLED
105	Lumped mass matrix	DYNAMICS
146	Include stiffness (second) Rayleigh damping parameter	DYNAMICS
275	Automatic time step calculation for implicit dynamics	DYNAMICS
254	Output all eigensolutions computed using inverse iteration	EIGENVALUE
311	Compute number of eigenvalues in specified range and terminate analysis	EIGENVALUE
93	Suppress intermediate eigenvalue output for Subspace or bracketing methods	EIGENVALUES
2	Suppress node coordinate checks for similar coordinates and missing nodes	INPUT
4	No overwriting present in data exit on first occurrence	INPUT
45	Suppress expanded input data printout for load cases	INPUT
47	Axisymmetry about the global X-axis	INPUT
48	Switch CBF input from Force/unit volume to acceleration	INPUT
49	Automatic correction of midside nodes	INPUT
51	Data processing only	INPUT
61	All 2D slideline surfaces defined with outward normals	INPUT
117	Invokes geometric assignments input	INPUT
118	Invokes material assignments input	INPUT
119	Invokes temperature input for joints (by default no temperature in joints)	INPUT
123	Clockwise node numbering	INPUT
207	Use angle in degrees to define direction of anisotropy for user-defined materials	INPUT
235	Heat fraction specified for thermal softening	INPUT
242	Temperatures input and output in degrees Celsius	INPUT
250	Datafile has been created by Modeller	INPUT
251	Data processing and create a restart file (linear static analyses only)	INPUT
253	Emissivity specified in thermal environment properties	INPUT
319	Invokes eccentricity input for QTS4 thick shell family	INPUT
324	Specify damping properties for joint properties general explicitly	INPUT
350	Invokes product moment of inertia (Iyz) input for BTS3 nonlinear thick beam elements	INPUT

No.	Effect of option (plus related notes where applicable)	Category
18	Invokes finer numerical integration rules for elements	INTEGRATION
19	Invokes coarse numerical integration rule for semiloof	INTEGRATION
20	Invokes non-orthogonal concrete crack model (concrete model 24)	INTEGRATION
22	Invokes fracture energy strain-softening model (concrete model 24)	INTEGRATION
23	Invokes strain-variable shear retention model (concrete model 24)	INTEGRATION
91	Formulate element mass with fine integration	INTEGRATION
134	Gauss to Newton-Cotes integration for elements	INTEGRATION
155	14 point integration rule for HX20	INTEGRATION
156	13 point integration rule for HX20	INTEGRATION
171	Use standard shear strain field for QTS4 thick shell elements	INTEGRATION
172	Formulate modulus matrix by integrating across the cross-section	INTEGRATION
225	Use alternative number of parameters for enhanced strain interpolation function	INTEGRATION
266	Layer by layer computation of mass matrix for solid composites	INTEGRATION
395	Use 14-point fine integration rule for mass matrix of TH10 family (used together with 91)	INTEGRATION
396	Improved top/middle/bottom transverse shear stress calculation for thick shell elements	INTEGRATION
398	Use all 27/18 integration points for stress extrapolation for 20/16 node solids with fine integration	INTEGRATION
102	Switch off load correction stiffness matrix due to centripetal acceleration	LOADING
202	Apply CLs in local cylindrical coordinates with Fourier elements	LOADING
381	Apply CBF,ELDS,UDL loads along beam axes with BMS3 elements	LOADING
385	Preserve loading assigned to deactivated elements for subsequent application on activation	LOADING
17	Keep global matrix assembly in-core for fast/iterative solvers	NONLINEAR
36	Follower loads (available for selected elements)	NONLINEAR
54	Updated Lagrangian geometric nonlinearity	NONLINEAR
62	Continue solution if more than one negative pivot occurs	NONLINEAR
87	Total Lagrangian geometric nonlinearity	NONLINEAR
138	Output yield flags only	NONLINEAR
139	Output yielded, cracked or crushed integration points only for MNL	NONLINEAR

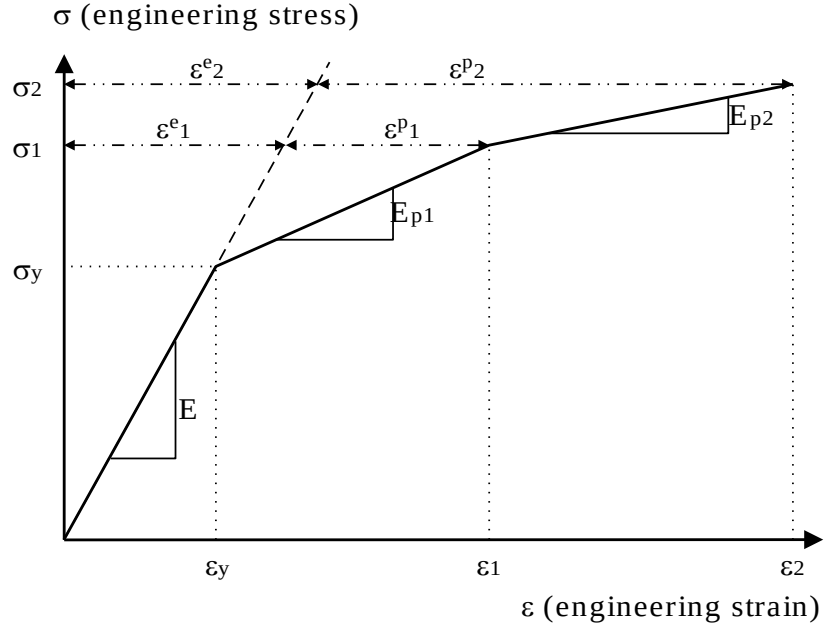
No.	Effect of option (plus related notes where applicable)	Category
157	Material Model 29 required (mandatory with non-cross-section elements)	NONLINEAR
164	Guide arc-length solution with current stiffness parameter	NONLINEAR
167	Eulerian geometric nonlinearity	NONLINEAR
179	Verify arguments to user-supplied constitutive model routines	NONLINEAR
187	Include all variables in convergence norm computations	NONLINEAR
229	Co-rotational nonlinear geometric formulation	NONLINEAR
261	Switch for selecting root with lowest residual norm (arc-length method)	NONLINEAR
273	Contact penalty stiffness update for augmented Lagrangian and slidelines	NONLINEAR
308	Arc-length solution using local relative displacements (interface elements)	NONLINEAR
340	Allow application of angular velocities with nonlinear auto incrementation	NONLINEAR
370	Strain hardening approach to modified von Mises and Hoffman material models (77/78)	NONLINEAR
16	Continue nonlinear/eigenvalue solution when failed to converge	NONLINEAR / EIGENVALUES
30	Output final node coordinates in ascending order	OPTIMISATION
72	Suppress machine code inner loops	OPTIMISATION
100	Output optimum frontal solution order	OPTIMISATION
170	Suppress storage of shapes	OPTIMISATION
197	Fast element stiffness formulation without sparsity checks	OPTIMISATION
230	Suppress stress calculation for eigenvalue analysis	OPTIMISATION
26	Reduce number of lines output in coordinates	OUTPUT
32	Suppresses stress output but not stress resultant output	OUTPUT
33	Output direction cosines of local cartesian systems for interface	OUTPUT
34	Output element stress resultants	OUTPUT
39	Stress smoothing for rubber material models	OUTPUT
40	Output nodal displacement increments for nonlinear analyses	OUTPUT
42	Output nodal residual forces (nonlinear)	OUTPUT
44	Suppress expanded input data printout except load cases	OUTPUT
46	Suppress page skip between output stages	OUTPUT
55	Output strains as well as stresses	OUTPUT
59	Output local direction cosines for shells	OUTPUT
77	Output principal stresses and directions for solids	OUTPUT
115	Output displacements and reactions in transformed axes	OUTPUT
116	Suppress the output of internal constraint forces	OUTPUT
143	Output shear forces in plate bending elements.	OUTPUT
144	Output element results for each load case separately	OUTPUT

No.	Effect of option (plus related notes where applicable)	Category
147	Omit output phase	OUTPUT
169	Suppress extrapolation of stress to nodes for semiloof shell	OUTPUT
181	Harmonic response/complex mode output required in polar system	OUTPUT
208	Write plot file in pre version 11.0 format (not valid for explicit dynamics)	OUTPUT
255	Output all radiation surface view factors to FACET file.	OUTPUT
259	Save nodal stresses in plot file to avoid stress extrapolation in Modeller	OUTPUT
290	Write element stiffness and mass matrices to plot file	OUTPUT
304	Plot/restart databases open until end of analysis. Speeds subsequent dumps.	OUTPUT
310	Output results to user-defined output file	OUTPUT
320	Global matrix (and righthand side) assembly and output only	OUTPUT
342	Always output progress information during solution	OUTPUT
345	Separate plot file for each analysis step	OUTPUT
348	Slideline summary output	OUTPUT
363	Suppress echo of computed laminate properties in output file (woven fabric material)	OUTPUT
364	Write element stiffness, mass and damping matrices to plot file	OUTPUT
373	Extended input data diagostics for CEB-FIP creep and shrinkage model	OUTPUT
380	Output stress resultants relative to beam axes for eccentric BTS3 elements	OUTPUT
389	Echo thermal link elements to output file	OUTPUT
390	Include selected results in plot file for data-only-processing	OUTPUT
14	Compute element stresses and strains (plates/flat shells)	PROCESSING
53	Positive definite/singularity check on modulus matrix at each Gauss point	PROCESSING
66	Calculate element internal forces	PROCESSING
110	Use assumed shear strain field for QTS8 thick shell elements	PROCESSING
111	Don't include superelement generation load	PROCESSING
136	Stress resultants by f=Kd and cubic interpolation	PROCESSING
137	Stress resultants by f=Kd and equilibrium	PROCESSING
227	Error if a value is outside the bounds of a table of values	PROCESSING
231	Error for material direction cosines not in plane of semiloof elements	PROCESSING
272	Activate or deactivate elements	PROCESSING
274	Use Uzawa rather than Powell method for node activity	PROCESSING

No.	Effect of option (plus related notes where applicable)	Category
278	Assign six degrees of freedom to all thick shell element nodes	PROCESSING
281	Autoloader file	PROCESSING
303	Exclude incompatible modes for solid composite elements	PROCESSING
318	Adaptive Analysis: Overwrite disp's with interpolated disp's from old mesh	PROCESSING
346	Single segment contact from beginning of problem	PROCESSING
347	Single segment contact from beginning of second increment	PROCESSING
387	Redefine NSET to angle in element xy-plane for use in orthotropic material properties	PROCESSING
388	Use duplicate nodes facility for collapsed elements	PROCESSING
394	Lamina directions supported	PROCESSING
397	With data-only-processing, compute memory requirements for solution of equations	PROCESSING
399	Retain inactive elements in birth and death solution (old analysis type)	PROCESSING
124	Save shape function array for restarts	RESTART
204	Use existing Plot file after restarting an analysis	RESTART
184	Suppress stringent slave search for slideline problems	SLIDELINES
185	Suppress initial slideline_surface stiffness check	SLIDELINES
186	Suppress initial penetration check for slidelines	SLIDELINES
352	Turn off consistently linearised formulation for slidelines	SLIDELINES
353	Use of a fixed penalty parameter with slidelines	SLIDELINES
355	Slideline Cohesion with Coulomb friction in 2D	SLIDELINES
386	Use of weighted force/stiffness distribution with slidelines	SLIDELINES
64	Non-symmetric frontal solution	SOLVER
131	Parallel frontal solver algorithm	SOLVER
247	Extra output for iterative solvers	SOLVER
248	Single precision preconditioning for PCG solver	SOLVER
252	Suppress pivot warning messages from frontal solution algorithm	SOLVER
279	Save the reduced stiffness for further use in a linear analysis	SOLVER
323	Extra preconditioning for iterative solver with a hierarchical basis	SOLVER
344	Suppress condition number estimate in fast solvers	SOLVER
376	Suppress use of METIS ordering for fast solvers	SOLVER
377	Use minimum amount of memory for BCS solvers	SOLVER

Appendix B Nonlinear Hardening Material Convention

Nonlinear Hardening Material Convention



Using the slope of the uniaxial yield stress against equivalent plastic strain:

$$C = \frac{E_p}{\left(1 - \frac{E_p}{E}\right)}$$

we have:

$$E_{p1} = \frac{\sigma_1 - \sigma_y}{\varepsilon_1 - \varepsilon_y} \qquad \text{and} \qquad E_{p2} = \frac{\sigma_2 - \sigma_1}{\varepsilon_2 - \varepsilon_1}$$

where:

$$\varepsilon_y = \frac{\sigma_y}{E} \qquad \varepsilon_1 = \varepsilon_1^e + \varepsilon_1^p \qquad \varepsilon_2 = \varepsilon_2^e + \varepsilon_2^p$$

which may be substituted into the top equation to give the corresponding C values for each section:

$$C = \frac{E_p}{\left(1 - \frac{E_p}{E}\right)} \qquad (Ep < E)$$

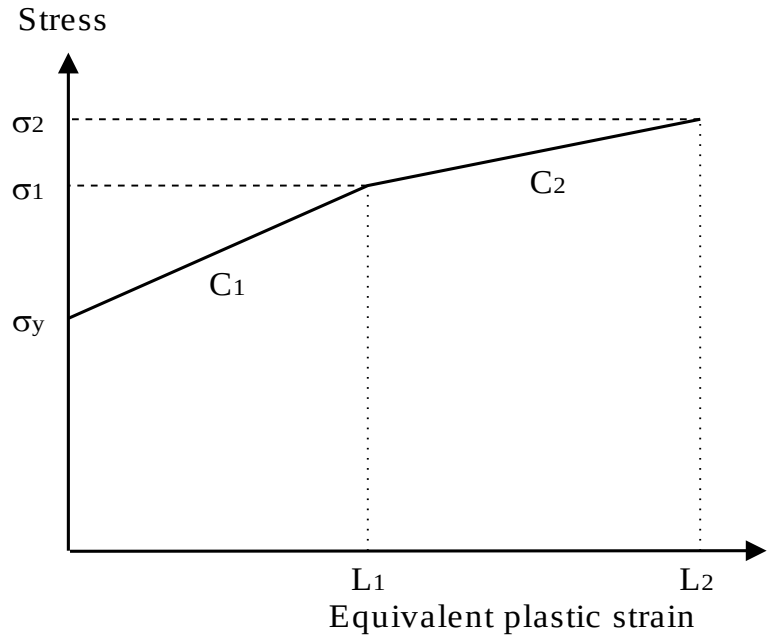
Now the strain values required by LUSAS are the ε_1^p, ε_2^p, etc. The limit on the equivalent plastic strain up to which the hardening curve is valid are, thus:

$$L_1 = \varepsilon_1 - \varepsilon_1^e = \varepsilon_1 - \frac{\sigma_1}{E}$$

and

$$L_2 = \varepsilon_2 - \varepsilon_2^e = \varepsilon_2 - \frac{\sigma_2}{E}$$

The converted curve for LUSAS use would, therefore, be as follows:


Stress
σ2
σ1
σy
C1
C2
L1
L2
Equivalent plastic strain

Appendix C Material Model Interface

Solver User Interface Routines

A number of interface facilities are available within the LUSAS Solver. Since the specification of these facilities requires both external development of FORTRAN source code and access to the Solver object libraries, these facilities are aimed at the advanced user.

The object libraries and user-interface routines needed for external code development are provided in the LUSAS MMI kit. The kit also includes workspace and project files that allow the development of external code in a visual environment on a PC, and which enable the compilation and linking of that code into Lusas Solver. Please contact your local distributor for further details.

User Defined Constitutive Models

For continuum based models the user is required to define three subroutines in FORTRAN to carry out the following tasks:

- `USRKDM` defines the modulus matrix.
- `USRSTR` defines the current stress and material state and state variables.
- `USRSVB` outputs evaluated non-linear state variables.

For resultant based models the user is required to define the following subroutines:

- `USRRDM` defines the modulus matrix.
- `USRRST` defines the current stress and material state and state variables.
- `USRSVB` outputs evaluated non-linear state variables.

See *Material Properties Nonlinear User* and *Material Properties Nonlinear Resultant User* for data syntax details.

User Defined Creep Models

This facility allows the specification of a creep law for a particular material if the creep models available in LUSAS Solver are inappropriate. The user-supplied subroutine

USRCRP allows specification of creep laws that are a function of stress, strain and temperature history.

See *Creep Properties User* for data syntax details.

User Defined Damage Models

The user-supplied damage subroutine **USRDAM** permits external computation of the damage variable and its derivative with respect to the current elastic complementary energy norm.

See *Damage Properties User* for data syntax details.

User Defined Viscoelastic Models

This facility permits an externally developed viscoelastic material model to be used within Lusas. The user-supplied viscoelastic subroutine **USRDMV** is used to compute the viscoelastic contribution to the modulus matrix.

See *Visco Elastic Properties User* for data syntax details.

User Defined Friction Models

The user-supplied subroutine **USRSLF** permits a non-linear friction law to be utilised in a slideline analysis. The friction law may be a function of the surface temperatures, the relative velocities and/or accelerations of adjacent surfaces and a set of user-defined friction parameters.

See *Slideline Properties User* for data syntax details.

User Defined Rate of Internal Heat Generation

The user-supplied subroutine **USRRHG** permits the way in which internal heat is generated in a thermal analysis to be defined. This can be defined to be a function of temperature, time and chemical reaction. The parameters used to control the chemical reaction are specified under the RIHG USER data chapter. In a thermo-mechanical coupled analysis, variables defining the degree and rate of chemical reaction (or cure) may be transferred to the structural analysis where they can be accessed in the user interface routines **`USRKDM`** and **`USRSTR`**. The modulus matrix and stress computations may then become a function of degree or rate of cure.

See *RIHG User* for data syntax details.

Software Required

A compatible FORTRAN compiler is required to include user-defined subroutines. Call your LUSAS distributor for details.

Programming Rules

Modification of External Arguments

It is important that only those arguments indicated in the user-programmable routines are modified. All other arguments may be manipulated as required, but must remain unchanged.

Although LUSAS contains a large number of system error traps, specifically designed to detect internally corrupted variables and artificially terminate the analysis procedure in a controlled manner, this process cannot generally be guaranteed. Hence, illegal modification of arguments may lead to unpredictable analysis termination or corrupt solutions. LUSAS can accept no responsibility in such circumstances.

For the same reason it is important that the DIMENSION, CHARACTER and LOGICAL declarations within the user-programmable subroutines are not modified or deleted. Only code between the lines indicated may be modified externally. These subroutines are described in detail in the following chapters.

Verification

The incoming and outgoing arguments to the user-programmable subroutines may be verified via lower level subroutines that are also provided. Access to the verification subroutines is activated by specification of Option 179 in the analysis data file. The arguments to these routines should not be altered.

The logical variable FEA is for LUSAS/FEA use only. These verification routines are described in detail in the following chapters.

External Error Diagnosis

The logical error flag (ERROR) is initialised to .FALSE., and may be used to detect a fatal error within the externally supplied FORTRAN code. On exit from the user-programmable subroutines detection of ERROR=.TRUE. will activate a controlled termination of the analysis procedure.

Argument Definition Codes

In the descriptions of all subroutines in the sections that follow, abbreviations are used to differentiate between variables and arrays for integer, real logical and character variables. These definitions are shown in the table right.

Code	Variable type	Code	Array type
IV	integer variable	IA	integer array
RV	real variable	RA	real array
LV	logical variable	LA	logical array
CV	character variable	CA	character array

Declaration

FEA Ltd can accept no responsibility whatsoever for any analysis or programming results obtained through use of the user-programmable CONSTITUTIVE MODELS, CREEP, DAMAGE, NONLINEAR FRICTION or MODELLER RESULTS. Whilst FEA Ltd can validate that the arguments are passed correctly through the interfaces to the user-programmable subroutines, FEA Ltd cannot be held responsible for any programming or alterations carried out by the user within these routines.

FEA Ltd cannot guarantee that the interfaces to the above-named user routines will continue to remain in the form described in this manual.

User Defined Constitutive Models

Continuum models

All constitutive models defined under MATERIAL PROPERTIES NONLINEAR USER utilise continuum stresses and strains and the user defined routines interface with code at the material integration point level. For continuum based elements (for example, bars, 2D and 3D continuum) this is within a Gauss point loop. For resultant based elements (for example semi-loof beams and shells) this is at a fibre or layer sampling position within the Gauss point loop.

The constitutive relationship is assumed to be of the form:

$$\{\Delta\sigma\} = \mathrm{D}\,\{\Delta\varepsilon\}$$

where $\{\Delta\sigma\}$ are the increments of continuum stress, $\{\Delta\varepsilon\}$ are the increments of continuum strain and D is the constitutive or modulus matrix. The modulus matrix is explicitly defined by the user via the externally developed FORTRAN subroutine, USRKDM, and is of the form:

$$\mathrm{D} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & \cdots & D_{1,\mathrm{ndse}} \\ D_{21} & D_{22} & D_{23} & \cdots & D_{2,\mathrm{ndse}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_{\mathrm{ndse},1} & D_{\mathrm{ndse},2} & D_{\mathrm{ndse},3} & \cdots & D_{\mathrm{ndse},\mathrm{ndse}} \end{bmatrix}$$

where **ndse** is the number of continuum stresses or strains at a material sampling point. The number of continuum stresses or strains at a point is related to the LUSAS model number **mdl**, and is a constant for each element type.

The continuum stress and strain components, and their associated model number (mdl) are tabulated for each applicable LUSAS element group below:

mdl	Model type	Components	ndse
1	Uniaxial	xx	1
2	Plane stress	xx, yy, xy	3
2	Semi-loof shell	xx, yy, xy	3
3	Plane strain (approximate)	xx, yy, xy	3
4	Axisymmetric	xx, yy, xy, zz	4
5	3D beams	xx, xy, xz	3
6	Solids	xx, yy, zz, xy, yz, zx	6
7	3D semi-loof beams	xx, xy, xz, yz	4
8	Axisymmetric sheet and shell	xx, yy	2
9	Axisymmetric thick shell	xx, xy, zz	3
10	Thick plane beam	xx, xy	2
11	Plane strain	xx, yy, xy, zz	4
12	Thick shells	xx, yy, xy, yz, zx	5

User Material Properties Input

The user material properties are input in a similar manner as the other LUSAS material types.

The user material input consists of a total of **nprz** material parameters, the first 15 of which are specifically for LUSAS use. These 15 properties are required should the user wish to utilise some of the other LUSAS analysis types (e.g. thermal or dynamic analyses) for which material parameters are required. The temperature is specified should the user wish to use temperature dependent properties. Material properties 16 to **nprz** must be supplied by the user in the order required by the user-supplied routines. It should be noted however, that all **nprz** properties can be used within the user routines. These properties, relating to the current temperature, are stored in the array ELPR. Values for the complete table of reference temperatures are stored in array ELPRT.

The 15 specific properties are:

1 Young's modulus (E)

2 Poisson's ratio (ν)

3 Mass density (ρ)

4-9 Coefficients of thermal expansion ($\alpha_x,\alpha_y,\alpha_z,\alpha_{xy},\alpha_{yz},\alpha_{xz}$)

10 Mass Rayleigh damping parameter (a_r)

11 Stiffness Rayleigh damping parameter (b_r)

12 Heat fraction (h_f)

13 Reference temperature (T)

14 Angle of anisotropy (θ) measured in degrees relative to the reference axes (with Option 207 set), or the Cartesian set number (nset) defining the local reference axes.

15 Not used at present.

Together with the user material properties the user also needs to specify the following integer numbers for the specific material defined:

- **lptusr** a number which identifies the particular user material model.
- **nprz** the total number of material properties used.
- **nstat** the number of nonlinear state variables that are used in the material model (these variables will be output together with the Gauss point stresses/strains). The value of **nstat** must not be less than 1.

The numbers are specified on the data input line for the user material model as follows:

```
MATERIAL PROPERTIES NONLINEAR USER  lptuser  nprz
nstat
```

See the *User-Supplied Nonlinear Material Properties* section for more details.

Evaluation of the Modulus Matrix

The modulus matrix D is explicitly defined via the externally supplied FORTRAN subroutine USRKDM. The routine is called at the material integration point level, from both the LUSAS Solver pre-solution and post-solution analysis modules. On entry to USRKDM, the modulus matrix D is fully initialised (each array component set to floating point real zeros). Hence, only the non-zero components of the modulus matrix need be evaluated. The returned modulus matrix must be symmetrical about the leading diagonal.

Nonlinear Stress Recovery

The stress recovery algorithm is defined explicitly via the externally supplied FORTRAN subroutine USRSTR, and is concerned with the evaluation of:

- **Stress State** the current stress state,
- **Material State** the current material state (as indicated by the nonlinear state variables),
- **Nonlinear Variables** additional associated nonlinear variables.

The routine is called at the material integration point level, from the LUSAS Solver post-solution analysis module. Stresses, strains and nonlinear state variables are available as current values, values at the end of the previous iteration, and values at the start of the current increment.

The procedure for updating incremental and iterative variables from the evaluated current values is automatically performed by LUSAS Solver. Consequently, modifications should be restricted to **current** values only.

Stress/Strain Formulation

For the Eulerian formulation and Green-Naghdi rate formulation, the stress components have been rotated to the initial configuration using the rotation matrix evaluated using polar decomposition. The stress rotation matrix is therefore that of the current configuration.

For the Jaumann rate formulation, the stress components at time t have been rotated to the current configuration. The stress rotation matrix is then the incremental spin matrix. In addition, no total strain is evaluated for the Jaumann rate.

Nonlinear State Variable Output

The output of the evaluated nonlinear state variables is controlled via the externally supplied FORTRAN subroutine, USRSVB. The routine is called at the material integration point level, from the LUSAS Solver output analysis module.

Default nonlinear state variable output for user supplied constitutive models is of the form shown below (6 variables to a line, to a total of **nstat** values. Nonlinear state variables for user-supplied models are output by default and are not subject to LUSAS Solver Options 138 and 139.

```
NL STATE VARIABLES     0.0000E+00 0.0000E+00 0.0000E+00
(contd.)
                       0.0000E+00 0.0000E+00 0.0000E+00
```

The output of stresses, and optionally strains, follows the normal pattern for the particular element type, i.e. it is subject to the usual LUSAS frequency and location controls. All output must be written to the recognised LUSAS output channel defined by integer NT6.

Verification

Three routines USRVF1, USRVF2 and USRVF3 are supplied to enable the user to verify the incoming and outgoing arguments to the user-programmed subroutines USRKDM, USRSTR and USRSVB respectively. These subroutines may be utilised by the user but no alterations are permitted.

SUBROUTINE USRKDM

Purpose

Explicit definition of the modulus matrix D for user-defined material. This routine may be programmed by the user but the argument list must not be altered in any way.

Note. For static nonlinear analysis failure to code this routine will result in slower convergence. For transient dynamic analysis however it is important that this routine defines the nonlinear modulus matrix as failure to do so may result in incorrect results.

```
      SUBROUTINE USRKDM
     A(    LPTUSR  ,MDL     ,NAXES   ,NDSE    ,NEL     ,
     B     NG      ,NPRZ    ,NSTAT   ,NT6     ,NTAB    ,
     C     NACTVE  ,DINT    ,DT      ,TEMPC   ,RSPTM   ,
     D     DOCT    ,DOCC    ,RDOC    ,TEMPR   ,VERIF   ,
     E     NWINC   ,REPAS   ,ERROR   ,INIACT  ,FEA     ,
     F     ELPR    ,EPST    ,EPSC    ,STVBT   ,STVBC   ,
     G     STRST   ,STRSC   ,XYZ     ,D       ,ELPRT   ,
     H     DRCMGP  )
      DIMENSION
     A     ELPR(NPRZ)     ,EPST(NDSE)     ,EPSC(NDSE)     ,
     B     STVBT(NSTAT)   ,STVBC(NSTAT)   ,STRST(NDSE)    ,
     C     STRSC(NDSE)    ,XYZ(NAXES)     ,D(NDSE,NDSE)   ,
     D     ELPRT(NPRZ,NTAB),DRCMGP(NAXES,NAXES)
```

Name	Argument Description	Type	Modify
LPTUSR	User supplied constitutive model reference number	IV	-
MDL	LUSAS reference code for stress model type	IV	-

Name	Argument Description	Type	Modify
NAXES	Number of system axes (dimensions)	IV	-
NDSE	Number of continuum stress components at a Gauss point for the stress model type MDL	IV	-
NEL	Current element number	IV	-
NG	Current Gauss point number	IV	-
NPRZ	Number of material parameters for the constitutive model	IV	-
NSTAT	Number of state dependent variables	IV	-
NT6	LUSAS results output channel	IV	-
NTAB	Number of temperature dependent property tables	IV	-
NACTVE	Element activation status 0-Standard element 1-activating 2-deactivated	IV	
DINT	Characteristic length or area of the current Gauss point	RV	-
DT	Current time step increment (dynamic analysis)	RV	-
TEMPC	Current temperature	RV	-
RSPTM	Current total response time (dynamic analysis)	RV	-
DOCT	Degree of cure at the start of the current increment (thermo-mechanical coupled analysis via user defined interface USRRHG)	RV	
DOCC	Current degree of cure (thermo-mechanical coupled analysis via user defined interface USRRHG)	RV	
RDOC	Current rate of cure (thermo-mechanical coupled analysis via user defined interface USRRHG)	RV	
TEMPR	Reference or initial temperature	RV	
VERIF	Logical incoming/outgoing argument verification flag	LV	-
NWINC	Logical flag, TRUE denoting the start of a new increment	LV	-
REPAS	Logical flag, TRUE denoting a re-pass	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
INIACT	Logical flag for initial activation of element	LV	
FEA	Logical flag for FEA use only	LV	-
ELPR	Material properties for current element (evaluated at the current temperature)	RA	-
EPST	Total strain components at the start of the current increment	RA	-
EPSC	Current total strain components	RA	-

Name	Argument Description	Type	Modify
STVBT	State variables at the start of the current increment	RA	-
STVBC	Current state variables	RA	-
STRST	Stress components at the start of the current increment	RA	-
STRSC	Current stress components	RA	-
XYZ	Coordinates for the current Gauss point	RA	-
D	Modulus matrix (D-matrix)	RA	yes
ELPRT	Table of temperature dependent material properties	RA	-
DRCMGP	Direction cosine matrix for the material	RA	-

SUBROUTINE USRSTR

Purpose

Stress recovery algorithm for user-defined material. This subroutine may be programmed by the user but the argument list may not be altered in any way.

Note. For a static nonlinear analysis, failure to code routine USRKDM will result in slower convergence. For a transient dynamic analysis, however, it is important that the routine USRKDM defines the nonlinear modulus matrix as failure to do so may result in incorrect results.

```
 SUBROUTINE USRSTR
A(    IMAT     ,LPTUSR   ,MDL      ,NAXES    ,NDSE     ,
B     NEL      ,NG       ,NPRZ     ,NSTAT    ,NT6      ,
C     NTAB     ,DINT     ,DISEN    ,DT       ,STREN    ,
D     TEMPC    ,RSPTM    ,VIHG     ,DOCT     ,DOCC     ,
E     RDOC     ,VERIF    ,STHLV    ,REPAS    ,ERROR    ,
F     BOUNDS   ,NWINC    ,INIACT   ,FEA      ,
G     ELPR     ,EPST     ,EPSI     ,EPSC     ,STVBT    ,
H     STVBI    ,STVBC    ,STRST    ,STRSI    ,STRSC    ,
I     XYZ      ,D        ,ELPRT    ,ROTAC    ,ROTAT    ,
J     DRCMGP   )
 DIMENSION
A     ELPR(NPRZ)      ,EPST(NDSE)      ,EPSI(NDSE)      ,
B     EPSC(NDSE)      ,STVBT(NSTAT)    ,STVBI(NSTAT)    ,
```

```
C     STVBC(NSTAT)   ,STRST(NDSE)    ,STRSI(NDSE)    ,
D     STRSC(NDSE)    ,XYZ(NAXES)     ,D(NDSE,NDSE)   ,
E     ELPRT(NPRZ,NTAB)       ,ROTAC(NAXES,NAXES)     ,
F     ROTAT(NAXES,NAXES)     ,DRCMGP(NAXES,NAXES)
```

Name	Argument Description	Type	Modify
IMAT	Material assignment reference number	IV	-
LPTUSR	User supplied constitutive model reference number	IV	-
MDL	LUSAS reference code for stress model type	IV	-
NAXES	Number of system axes (dimensions)	IV	-
NDSE	Number of continuum stress components at a Gauss point for the stress model type MDL	IV	-
NEL	Current element number	IV	-
NG	Current Gauss point number	IV	-
NPRZ	Number of material parameters for the constitutive model	IV	-
NSTAT	Number of state dependent variables	IV	-
NT6	LUSAS results output channel	IV	-
NTAB	Number of temperature dependent property tables	IV	-
DINT	Characteristic length or area of the current Gauss point	RV	-
DISEN	Energy dissipated by inelastic processes	RV	yes
DT	Current time step increment (dynamic analysis)	RV	-
STREN	Increment of elastic strain energy	RV	yes
TEMPC	Current temperature	RV	-
RSPTM	Current total response time (dynamic analysis)	RV	-
VIHG	Volumetric internal heat generation due to mechanical work	RV	yes
DOCT	Degree of cure at the start of the current increment (thermo-mechanical coupled analysis via user defined interface USRRHG)	RV	
DOCC	Current degree of cure (thermo-mechanical coupled analysis via user defined interface USRRHG)	RV	
RDOC	Current rate of cure (thermo-mechanical coupled analysis via user defined interface USRRHG)	RV	

Name	Argument Description	Type	Modify
VERIF	Logical incoming/outgoing argument verification flag	LV	-
STHLV	Logical increment step halving flag	LV	yes
REPAS	Logical flag for re-pass through subroutine	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
BOUNDS	Flag to error if temperature out of bounds of table	LV	-
NWINC	Logical flag, TRUE denoting the start of a new increment	LV	
INIACT	Logical flag for initial activation of element	LV	
FEA	Logical flag for FEA use only	LV	-
ELPR	Material properties for current element (evaluated at the current temperature	RA	-
EPST	Total strain components at the start of current increment	RA	-
EPSI	Total strain components at the end of the previous iteration	RA	-
EPSC	Current total strain components	RA	-
STVBT	State variables at the start of the current increment	RA	-
STVBI	State variables at the end of the previous iteration	RA	-
STVBC	On entry: State variables at start of the current increment, on exit: Current state variables	RA	yes
STRST	Stress components at the start of the current increment	RA	-
STRSI	Stress components at the end of the previous iteration	RA	-
STRSC	On entry: Stress components at start of current increment. On exit: Current stress components	RA	yes
XYZ	Coordinates of the current Gauss point	RA	-
D	Modulus matrix (D-matrix)	RA	yes
ELPRT	Table of temperature dependent material properties	RA	-
ROTAC	Stress rotation matrix (current)	RA	-
ROTAT	Stress rotation matrix (at time t)	RA	-
DRCMGP	Direction cosine matrix for the material	RA	-

SUBROUTINE USRSVB

Purpose

Output of nonlinear state variables for user-defined material. This subroutine may be programmed by the user but the argument list must not be altered in any way.

```
 SUBROUTINE USRSVB
A(    LPTUSR  ,NSTAT   ,NDSE    ,NT6     ,VERIF   ,
B     ERROR   ,FEA     ,
C     STVBC   )
 DIMENSION
A     STVBC(NSTAT)
```

Name	Argument Description	Type	Modify
LPTUSR	User supplied constitutive model reference number	IV	-
NSTAT	Number of state dependent variables	IV	-
NDSE	Number of continuum stress components at a Gauss point for the stress model type MDL	IV	-
NT6	LUSAS results output channel	IV	-
VERIF	Logical incoming/outgoing argument verification flag	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
FEA	Logical flag for FEA use only	LV	-
STVBC	Current state variables for constitutive model	RA	-

SUBROUTINE USRVF1

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutine USRKDM. This subroutine may be utilised by the user but no alterations are permitted.

```
 SUBROUTINE USRVF1
A(    LPTUSR  ,MDL     ,NAXES   ,NDSE    ,NEL     ,
B     NG      ,NPRZ    ,NSTAT   ,NT6     ,NTAB    ,
C     NACTVE  ,DINT    ,DT      ,TEMPC   ,RSPTM   ,
D     DOCT    ,DOCC    ,RDOC    ,TEMPR   ,VERIF   ,
E     NWINC   ,REPAS   ,ERROR   ,INIACT  ,FEA     ,
```

```
F     ARGIN    ,
G     ELPR     ,EPST     ,EPSC     ,STVBT    ,STVBC    ,
H     STRST    ,STRSC    ,XYZ      ,D        ,ELPRT    ,
I     DRCMGP   )
 DIMENSION
A     ELPR(NPRZ)      ,EPST(NDSE)      ,EPSC(NDSE)      ,
B     STVBT(NSTAT)    ,STVBC(NSTAT)    ,STRST(NDSE)     ,
C     STRSC(NDSE)     ,XYZ(NAXES)      ,D(NDSE,NDSE)    ,
D     ELPRT(NPRZ,NTAB),DRCMGP(NAXES,NAXES)
```

The argument list is the same as that for subroutine USRKDM, except for one argument. The additional argument is the logical variable ARGIN (see continuation line F in the above subroutine statement) that is used in this subroutine. If ARGIN=.TRUE. verification of incoming arguments will be carried out and if ARGIN=.FALSE. verification of outgoing arguments will take place.

SUBROUTINE USRVF2

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutine USRSTR. This subroutine may be utilised by the user but no alterations are permitted.

```
 SUBROUTINE USRVF2
A(    IMAT     ,LPTUSR   ,MDL      ,NAXES    ,NDSE     ,
B     NEL      ,NG       ,NPRZ     ,NSTAT    ,NT6      ,
C     NTAB     ,DINT     ,DISEN    ,DT       ,STREN    ,
D     TEMPC    ,RSPTM    ,VIHG     ,DOCT     ,DOCC     ,
E     RDOC     ,VERIF    ,STHLV    ,REPAS    ,ERROR    ,
F     BOUNDS   ,NWINC    ,INIACT   ,FEA      ,ARGIN    ,
G     ELPR     ,EPST     ,EPSI     ,EPSC     ,STVBT    ,
H     STVBI    ,STVBC    ,STRST    ,STRSI    ,STRSC    ,
I     XYZ      ,D        ,ELPRT    ,ROTAC    ,ROTAT    ,
J     DRCMGP   )
 DIMENSION
A     ELPR(NPRZ)      ,EPST(NDSE)      ,EPSI(NDSE)      ,
B     EPSC(NDSE)      ,STVBT(NSTAT)    ,STVBI(NSTAT)    ,
C     STVBC(NSTAT)    ,STRST(NDSE)     ,STRSI(NDSE)     ,
D     STRSC(NDSE)     ,XYZ(NAXES)      ,D(NDSE,NDSE)    ,
```

```
E     ELPRT(NPRZ,NTAB)        ,ROTAC(NAXES,NAXES)     ,
F     ROTAT(NAXES,NAXES)      ,DRCMGP(NAXES,NAXES)
```

The argument list is the same as that for subroutine USRSTR, except for one argument. The additional argument is the logical variable ARGIN (see continuation line F in the above subroutine statement) that is used in this subroutine. If ARGIN=.TRUE. verification of incoming arguments will be carried out and if ARGIN=.FALSE. verification of outgoing arguments will take place.

SUBROUTINE USRVF3

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutine USRSVB. This subroutine may be utilised by the user but no alterations are permitted.

```
 SUBROUTINE USRVF3
A(    LPTUSR   ,NSTAT    ,NDSE     ,NT6      ,VERIF    ,
B     ERROR    ,FEA      ,ARGIN    ,
C     STVBC    )
 DIMENSION
A     STVBC(NSTAT)
```

The argument list is the same as that for subroutine USRSVB, except for one argument. The additional argument is the logical variable ARGIN (see continuation line B in the above subroutine statement) that is used in this subroutine. If ARGIN=.TRUE. verification of incoming arguments will be carried out and if ARGIN=.FALSE. verification of outgoing arguments will take place.

User Defined Resultant Models

Resultant Models

All constitutive models defined under MATERIAL PROPERTIES NONLINEAR RESULTANT USER utilise stress resultants and strains/curvatures. The user defined routines interface with code at the element gauss point level. This facility may be used with 2D and 3D beam elements which have a nonlinear capability, namely BM3, BMX3,BTS3, BSL3/4, BXL4, BS3/4 and BSX4

The constitutive relationship is assumed to be of the form:

$$\{\Delta S\} = D\,\{\Delta\varepsilon\}$$

where $\{\Delta S\}$ are the increments of stress resultants, $\{\Delta\varepsilon\}$ are the increments of strains and curvatures and D is the constitutive or modulus matrix. The modulus matrix is

explicitly defined by the user via the externally developed FORTRAN subroutine, USRRDM, and is of the form:

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & \cdots & D_{1,ndse} \\ D_{21} & D_{22} & D_{23} & \cdots & D_{2,ndse} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_{ndse,1} & D_{ndse,2} & D_{ndse,3} & \cdots & D_{ndse,ndse} \end{bmatrix}$$

where **ndse** is the number of stress resultants or strains and curvatures at an element gauss point. The number of stress resultants at a point can be ascertained from the LUSAS element type number **nelt**.

The stress resultant and strain components, and their associated element type number (nelt) are tabulated for each applicable LUSAS beam element below:

nelt	Element name	Stress resultants	ndse
54,43	BM3,BMX3	Fx,Mz	2
205	BTS3	Fx,Fy,Fz,Mx,My,Mz	6
97,170,171	BS3,BS4,BSX4	Fx,My,Mz,Txz,Txy,Fy	6
98,176,177	BSL3,BSL4,BXL4	Fx,My,Mz,Txz,Txy,Fy	6

Resultant User Material Properties Input

The user material properties are input in a similar manner to the other LUSAS material types.

The resultant user material input consists of a total of **nprz** material parameters, the first 10 of which are specifically for LUSAS use. These 10 properties are required should the user wish to utilise some of the other LUSAS analysis types (e.g. thermal or dynamic analyses) for which material parameters are required. The temperature is specified should the user wish to use temperature dependent properties. Material properties 11 to **nprz** must be supplied by the user in the order required by the user-supplied routines. It should be noted however, that all **nprz** properties can be used within the user routines. These properties, relating to the current temperature, are stored in the array ELPR. Values for the complete table of reference temperatures are stored in array ELPRT.

The 10 specific properties are:

1 Young's modulus (E)

2 Poisson's ratio (ν)

3 Mass density (ρ)

4 Coefficients of thermal expansion (α)

5 Mass Rayleigh damping parameter (ar)

6 Stiffness Rayleigh damping parameter (br)

7 Not used at present

8 Reference temperature (T)

9 Not used at present

10 Not used at present.

Together with the user material properties the user also needs to specify the following integer numbers for the specific material defined:

- **lptusr** a number which identifies the particular user material model.
- **nprz** the total number of material properties used.
- **ndcrve** the number of material data curves defined. This allows the user to input a table of values that vary with reference to something other than temperature. For example, a table of moment-curvature profiles may be defined where each profile relates to a particular axial force in the element.
- **nstat** the number of nonlinear state variables that are used in the material model (these variables will be output together with the Gauss point stresses/strains). The value of **nstat** must not be less than 1.

The numbers are specified on the data input line for the user material model as follows:

```
MATERIAL PROPERTIES NONLINEAR RESULTANT USER  lptuser
nprz  ndcrve  nstat
```

See the *User-Supplied Nonlinear Material Properties* section for more details.

Evaluation of the Modulus Matrix

The modulus matrix D is explicitly defined via the externally supplied FORTRAN subroutine USRRDM. The routine is called at the element gauss point level, from both the LUSAS Solver pre-solution and post-solution analysis modules. On entry to USRRDM, the modulus matrix D is fully initialised (each array component is set to a floating point real zero). Hence, only the non-zero components of the modulus matrix

need be evaluated. The returned modulus matrix must be symmetrical about the leading diagonal.

Nonlinear Stress Recovery

The stress recovery algorithm is defined explicitly via the externally supplied FORTRAN subroutine USRRST, and is concerned with the evaluation of:

- **Stress State** the current stress state,
- **Material State** the current material state (as indicated by the nonlinear state variables),
- **Nonlinear Variables** additional associated nonlinear variables.

The routine is called at the element gauss point level, from the LUSAS Solver post-solution analysis module. Stress resultants, strains and curvatures, and nonlinear state variables are available as current values, values at the end of the previous iteration, and values at the start of the current increment.

The procedure for updating incremental, and iterative variables from the evaluated current values, is automatically performed by LUSAS Solver. Consequently, modifications should be restricted to **current** values only.

Nonlinear State Variable Output

The output of the evaluated nonlinear state variables is controlled via the externally supplied FORTRAN subroutine, USRSVB. The routine is called at the element gauss point level from the LUSAS output analysis module.

Default nonlinear state variable output for user supplied constitutive models is of the form shown below (6 variables to a line, to a total of **nstat** values). Nonlinear state variables for user-supplied models are output by default and are not subject to LUSAS Solver Options 138 and 139.

```
NL STATE VARIABLES     0.0000E+00 0.0000E+00 0.0000E+00
(contd.)
                       0.0000E+00 0.0000E+00 0.0000E+00
```

The output of stress resultants, and optionally strains and curvatures, follows the normal pattern for the particular element type, i.e. it is subject to the usual LUSAS frequency and location controls. All output must be written to the recognised LUSAS output channel defined by integer NT6.

Verification

Two routines USRVF4 and USRVF3 are supplied to enable the user to verify the incoming and outgoing arguments to the user-programmed subroutines USRRDM and USRRST, and USRSVB respectively. This subroutine may be utilised by the user but no alterations are permitted.

SUBROUTINE USRRDM

Purpose

Explicit definition of the modulus matrix D for user-defined material. This routine may be programmed by the user but the argument list must not be altered in any way.

```
 SUBROUTINE USRRDM
A(    NEL     ,NELT    ,NDSE    ,NGP     ,NG      ,
B     LPTUSR  ,NDCRVE  ,NPRZ    ,NTAB    ,NLGPR   ,
C     NSTAT   ,NT6     ,IMAT    ,INC     ,ELEN    ,
D     DLENTH  ,DT      ,TEMPC   ,RSPTM   ,ELTAGE  ,
E     ACTIM   ,VERIF   ,NWINC   ,REPAS   ,ERROR   ,
F     FEA     ,BOUNDS  ,
G     ELGPR   ,ELPR    ,EPST    ,EPSC    ,STRST   ,
H     STRSC   ,STVBT   ,STVBC   ,PLSPRP  ,STRSI   ,
I     STRSR   ,EPSTH   ,EPSI    ,DUMMY   ,ELPRT   ,
J     SHRNKT  ,SHRNKC  ,D       )
 DIMENSION
A     ELGPR(NLGPR)   ,ELPR(NPRZ)     ,EPST(NDSE)     ,
B     EPSC(NDSE)     ,STRST(NDSE)    ,STRSC(NDSE)    ,
C     STVBT(NSTAT)   ,STVBC(NSTAT)   ,PLSPRP(NDSE)   ,
D     STRSI(NDSE)    ,STRSR(NDSE)    ,EPSTH(NDSE)    ,
E     EPSI(NDSE)     ,DUMMY(NDSE)  ,ELPRT(NPRZ,NTAB),
F     SHRNKT(NDSE)   ,SHRNKC(NDSE)   ,D(NDSE,NDSE)
```

Name	Argument Description	Type	Modify
NEL	Current element number	IV	-
NELT	Element type number	IV	-
NDSE	Number of stress components at a Gauss point	IV	-
NGP	Number of gauss points	IV	-
NG	Current Gauss point number	IV	-
LPTUSR	User supplied constitutive model reference number	IV	-
NDCRVE	Number of material data curves defined	IV	-
NPRZ	Number of material parameters for each material	IV	-
NTAB	Number of lines in a table of material properties	IV	-
NLGPR	Number of element geometric properties	IV	-

Name	Argument Description	Type	Modify
NSTAT	Number of state dependent model variables	IV	-
NT6	LUSAS results output channel	IV	-
IMAT	Material assignment number	IV	-
INC	Load increment number	IV	-
ELEN	Element length	RV	-
DLENGTH	Characteristic length of the current Gauss point	RV	-
DT	Current time step increment (dynamic analysis)	RV	-
TEMPC	Current temperature	RV	-
RSPTM	Current total response time (dynamic analysis)	RV	-
ELTAGE	Age of an element at activation	RV	
ACTIM	Time of element activation	RV	
VERIF	Logical incoming/outgoing argument verification flag	LV	-
NWINC	Logical flag, TRUE denoting the start of a new increment	LV	-
REPAS	Logical flag, TRUE denoting a re-pass	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
FEA	Logical flag denoting FEA or external code use (FOR FEA INTERNAL USE ONLY)	LV	-
BOUNDS	Flag to error if a value is outside the bounds of a table of values (OPTION 227)	LV	-
ELGPR	Element geometric properties	RA	-
ELPR	Material properties for current element (evaluated at the current temperature)	RA	-
EPST	Total strain components at the start of the current increment	RA	-
EPSC	Current total strain components	RA	-
STRST	Stress resultant components at the start of the current increment	RA	-
STRSC	Stress resultant components	RA	-
STVBT	State variables at the start of current increment	RA	-
STVBC	Current state variables	RA	-
PLSPRP	Vector of plastic geometric properties (defined using OPTION 157)	RA	-
STRSI	Initial stresses (these have already been added to the incoming STRSC/STRST values)	RA	-
STRSR	Residual stresses (these have already been added to the incoming STRSC/STRST values)	RA	-

Name	Argument Description	Type	Modify
EPSTH	Total thermal strains	RA	-
EPSI	Initial strains	RA	-
DUMMY	Unused vector	RA	-
ELPRT	Table of temperature dependent material properties	RA	-
SHRNKT	Shrinkage strain vector at start of current increment	RA	
SHRNKC	Current shrinkage strain vector	RA	
D	Modulus matrix (D-matrix)	RA	yes

SUBROUTINE USRRST

Purpose

Stress recovery algorithm for user-defined material. This subroutine may be programmed by the user but the argument list may not be altered in any way.

```
      SUBROUTINE USRRST
     A(    NEL      ,NELT     ,NDSE     ,NGP      ,NG       ,
     B     LPTUSR   ,NDCRVE   ,NPRZ     ,NTAB     ,NLGPR    ,
     C     NSTAT    ,NT6      ,IMAT     ,INC      ,ELEN     ,
     D     DLENTH   ,DT       ,TEMPC    ,RSPTM    ,ELTAGE   ,
     E     ACTIM    ,VERIF    ,NWINC    ,REPAS    ,ERROR    ,
     F     FEA      ,BOUNDS   ,
     G     ELGPR    ,ELPR     ,EPST     ,EPSC     ,STRST    ,
     H     STRSC    ,STVBT    ,STVBC    ,PLSPRP   ,STRSI    ,
     I     STRSR    ,EPSTH    ,EPSI     ,DEPS     ,ELPRT    ,
     J     SHRNKT   ,SHRNKC   ,D        )
      DIMENSION
     A     ELGPR(NLGPR)   ,ELPR(NPRZ)      ,EPST(NDSE)      ,
     B     EPSC(NDSE)     ,STRST(NDSE)     ,STRSC(NDSE)     ,
     C     STVBT(NSTAT)   ,STVBC(NSTAT)    ,PLSPRP(NDSE)    ,
     D     STRSI(NDSE)    ,STRSR(NDSE)     ,EPSTH(NDSE)     ,
     E     EPSI(NDSE)     ,DEPS(NDSE)    ,ELPRT(NPRZ,NTAB),
     F     SHRNKT(NDSE)   ,SHRNKC(NDSE)    ,D(NDSE,NDSE)
```

Name	Argument Description	Type	Modify
NEL	Current element number	IV	-
NELT	Element type number	IV	-
NDSE	Number of stress components at a Gauss point	IV	-
NGP	Number of gauss points	IV	-
NG	Current Gauss point number	IV	-
LPTUSR	User supplied constitutive model reference number	IV	-
NDCRVE	Number of material data curves defined	IV	-
NPRZ	Number of material parameters for each material	IV	-
NTAB	Number of lines in a table of material properties	IV	-
NLGPR	Number of element geometric properties	IV	-
NSTAT	Number of state dependent model variables	IV	-
NT6	LUSAS results output channel	IV	-
IMAT	Material assignment number	IV	-
INC	Load increment number	IV	-
ELEN	Element length	RV	-
DLENGTH	Characteristic length of the current Gauss point	RV	-
DT	Current time step increment (dynamic analysis)	RV	-
TEMPC	Current temperature	RV	-
RSPTM	Current total response time (dynamic analysis)	RV	-
ELTAGE	Age of an element at activation	RV	
ACTIM	Time of element activation	RV	
VERIF	Logical incoming/outgoing argument verification flag	LV	-
NWINC	Logical flag, TRUE denoting the start of a new increment	LV	-
REPAS	Logical flag, TRUE denoting a re-pass	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
FEA	Logical flag denoting FEA or external code use (FOR FEA INTERNAL USE ONLY)	LV	-
BOUNDS	Flag to error if a value is outside the bounds of a table of values (OPTION 227)	LV	-
ELGPR	Element geometric properties	RA	-
ELPR	Material properties for current element (evaluated at the current temperature)	RA	-
EPST	Total strain components at the start of the current increment	RA	-
EPSC	Current total strain components	RA	-

Name	Argument Description	Type	Modify
STRST	Stress resultant components at the start of the current increment	RA	-
STRSC	On entry: Stress resultant components at start of current increment. On exit: Current stress resultant components	RA	yes
STVBT	State variables at the start of current increment	RA	-
STVBC	On entry: State variables at start of the current increment, on exit: Current state variables	RA	yes
PLSPRP	Vector of plastic geometric properties (defined using OPTION 157)	RA	-
STRSI	Initial stresses (these have already been added to the incoming STRSC/STRST values)	RA	-
STRSR	Residual stresses (these have already been added to the incoming STRSC/STRST values)	RA	-
EPSTH	Total thermal strains	RA	-
EPSI	Initial strains	RA	-
DEPS	Incremental strains after deduction of initial and thermal strains	RA	-
ELPRT	Table of temperature dependent material properties	RA	-
SHRNKT	Shrinkage strain vector at start of current increment	RA	
SHRNKC	Current shrinkage strain vector	RA	
D	Modulus matrix (D-matrix)	RA	yes

SUBROUTINE USRVF4

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutines USRRDM and USRRST. This subroutine may be utilised by the user but no alterations are permitted.

```
      SUBROUTINE USRVF4
     A(    NEL     ,NELT    ,NDSE    ,NGP     ,NG      ,
     B     LPTUSR  ,NDCRVE  ,NPRZ    ,NTAB    ,NLGPR   ,
     C     NSTAT   ,NT6     ,IMAT    ,INC     ,ELEN    ,
     D     DLENTH  ,DT      ,TEMPC   ,RSPTM   ,ELTAGE  ,
     E     ACTIM   ,VERIF   ,NWINC   ,REPAS   ,ERROR   ,
     F     FEA     ,BOUNDS  ,ARGIN   ,RNAME   ,
     G     ELGPR   ,ELPR    ,EPST    ,EPSC    ,STRST   ,
```

```
H     STRSC    ,STVBT    ,STVBC    ,PLSPRP   ,STRSI    ,
I     STRSR    ,EPSTH    ,EPSI     ,DEPS     ,ELPRT    ,
J     SHRNKT   ,SHRNKC   ,D            )
 DIMENSION
A     ELGPR(NLGPR)   ,ELPR(NPRZ)      ,EPST(NDSE)     ,
B     EPSC(NDSE)     ,STRST(NDSE)     ,STRSC(NDSE)    ,
C     STVBT(NSTAT)   ,STVBC(NSTAT)    ,PLSPRP(NDSE)   ,
D     STRSI(NDSE)    ,STRSR(NDSE)     ,EPSTH(NDSE)    ,
E     EPSI(NDSE)     ,DEPS(NDSE)   ,ELPRT(NPRZ,NTAB),
F     SHRNKT(NDSE)   ,SHRNKC(NDSE)    ,D(NDSE,NDSE)
```

The argument list is the same as that for subroutine USRRST, except for the additional arguments ARGIN and RNAME. The logical variable ARGIN (see continuation line F in the above subroutine statement) is used in this subroutine. If ARGIN=.TRUE. verification of incoming arguments will be carried out and if ARGIN=.FALSE. verification of outgoing arguments will take place. The character variable RNAME indicates the routine for which verification is taking place, i.e. USRRDM or USRRST.

User Defined Joint Models

All joint models defined under JOINT PROPERTIES NONLINEAR USER utilise forces and strains/curvatures. The user defined routines interface with code at the element level. This facility may be used with any 2D or 3D joint element.

The constitutive relationship is assumed to be of the form:

$$\{\Delta F\} = [D]\{\Delta \varepsilon\}$$

where $\{\Delta F\}$ are the increments in joint forces, $\{\Delta \varepsilon\}$ are the increments of strains and curvatures and D is the diagonal modulus matrix. The modulus matrix is explicitly defined by the user via the externally developed FORTRAN subroutine, USRKJN, and is of the form:

$$[D] = \begin{bmatrix} D_{11} & 0 & 0 & \cdots & 0 \\ 0 & D_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D_{ndse,ndse} \end{bmatrix}$$

where **ndse** is the number of joint forces or strains and curvatures.

User Joint Properties Input

The user joint properties are input in a similar manner to the other LUSAS joint property types.

The user joint properties input consists of **nprz** parameters where **nprz** = **nprzs***N+**nprzj**+1. The value **nprzs** is the number of parameters per spring, **N** is the number of springs and **nprzj** is the number of parameters common to all springs. The properties $M_i, C_i, \alpha_i, a_{ri}, b_{ri}$ along with **mcode** are reserved for LUSAS internal use: they are only utilisd if other features of the program are required (e.g. dynamic and thermal analyses). It should be noted however, that all **nprz** properties can be used within the user routines.

Together with the user joint parameters the user also needs to specify the following integer numbers for the specific joint model defined:

- **lptusr** a number which identifies the particular user joint model.
- **nprzs** the number of input parameters per spring
- **N** the number of springs in the joint
- **nprzj** the number of input parameters common to each spring
- **nstat** the number of nonlinear state variables that are used in the joint model. The value of **nstat** must not be less than 1.

The numbers are specified on the data input line for the user material model as follows:

```
JOINT PROPERTIES NONLINEAR USER lptusr N nprzs nprzj nstat
```

Evaluation of the Modulus Matrix

The modulus matrix D is explicitly defined via the externally supplied FORTRAN subroutine USRKJN. The routine is called at the element level, from both the LUSAS Solver pre-solution and post-solution analysis modules. On entry to USRKJN, the modulus matrix D is fully initialised (each array component is set to a floating point real zero). Hence, only the non-zero components of the modulus matrix need be evaluated.

Nonlinear Stress Recovery

The stress recovery algorithm is defined explicitly via the externally supplied FORTRAN subroutine USRSJN, and is concerned with the evaluation of:

- **Stress State** the current stress state,
- **Material State** the current material state (as indicated by the nonlinear state variables),
- **Nonlinear Variables** additional associated nonlinear variables.

The routine is called at the element level, from the LUSAS Solver post-solution analysis module. Forces, strains and curvatures, and nonlinear state variables are available as current values, values at the end of the previous iteration, and values at the start of the current increment.

The procedure for updating incremental, and iterative variables from the evaluated current values, is automatically performed by LUSAS Solver. Consequently, modifications should be restricted to **current** values only.

Verification

Two routines USRJN1 and USRJN2 are supplied to enable the user to verify the incoming and outgoing arguments to the user-programmed subroutines USRKJN and USRSJN respectively.

SUBROUTINE USRKJN

Purpose

Explicit definition of the modulus matrix D for user-defined joint properties. This routine may be programmed by the user but the argument list must not be altered in any way.

```
      SUBROUTINE USRKJN
     A(    LNODZ   ,LPTUSR   ,NACTVE   ,NAXES    ,ND       ,
     B     NDSE    ,NEL      ,NPRZ     ,NSTAT    ,NT6      ,
     C     ACTIM   ,ELTAGE   ,DT       ,TEMPC    ,RSPTM    ,
     D     TEMPR   ,ERROR    ,FEA      ,INIACT   ,NWINC    ,
     E     REPAS   ,VERIF    ,
     F     B       ,D        ,DC       ,DISPT    ,DISPC    ,
     G     ELPR    ,EPST     ,EPSC     ,STVBT    ,STVBC    ,
     H     STRST   ,STRSC    ,XYZ      ,VELCTY   )
      DIMENSION
     A     B(NDSE,NDF)   ,D(NDSE,NDSE)   ,DC(NAXES,NAXES),
     B     DISPT(NDF)    ,DISPC(NDF)     ,ELPR(NPRZ)      ,
     C     EPST(NDSE)    ,EPSC(NDSE)     ,STVBT(NSTAT)    ,
     D     STVBC(NSTAT)  ,STRST(NDSE)    ,STRSC(NDSE)     ,
     E     XYZ(NAXES,LNODZ),VELCTY(NDF)
```

Name	Argument Description	Type	Modify
LNODZ	Number of nodes	IV	-
LPTUSR	User supplied constitutive model reference number	IV	-
NACTVE	Element activation status 0-Standard element 1-activating 2-deactivated	IV	-
NAXES	Number of system axes (dimensions)	IV	-
NDF	Number of degrees of freedom	IV	-
NDSE	Number of continuum stress components at a Gauss point for the stress model type MDL.	IV	-
NEL	Current element number	IV	-
NPRZ	Number of material parameters for the constitutive model	IV	-
NSTAT	Number of state dependent model variables	IV	-
NT6	LUSAS results output channel	IV	-
ACTIM	Element activation time	RV	-
ELTAGE	Element age	RV	-
DT	Current time step increment (dynamic analysis)	RV	-
TEMPC	Current temperature	RV	-
RSPTM	Current total response time (dynamic analysis)	RV	-
TEMPR	Reference temperature	RV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
FEA	Logical flag denoting FEA or external code use	LV	-
INIACT	Logical flag for initial activation of element	LV	-
NWINC	Logical flag, TRUE indicating the start of a new increment	LV	-
REPAS	Logical flag, TRUE for re-pass	LV	-
VERIF	Logical incoming argument verification flag	LV	-
B	Strain displacement matrix	RA	-
D	Modulus matrix (D-matrix)	RA	yes
DC	Direction cosine matrix for the local axes system	RA	-
DISPT	Array of displacements at the start of the current increment	RA	-
DISPC	Array of current displacements	RA	-
ELPR	Material properties for current element	RA	-
EPST	Total strain components at the start of the current increment	RA	-
EPSC	Current total strain components	RA	-

Name	Argument Description	Type	Modify
STVBT	State variables at the start of current increment	RA	-
STVBC	Current state variables	RA	-
STRST	Stress components at the start of the current increment	RA	-
STRSC	Stress components	RA	-
XYZ	Coordinates for the nodes	RA	-
VELCTY	Current velocity vector	RA	-

SUBROUTINE USRSJN

Purpose

Stress recovery algorithm for user-defined joint properties. This subroutine may be programmed by the user but the argument list may not be altered in any way.

```
      SUBROUTINE USRSJN
     A(    LNODZ   ,LPTUSR    ,NACTVE   ,NAXES    ,ND       ,
     B     NDSE    ,NEL       ,NPRZ     ,NSTAT    ,NT6      ,
     C     LSTATM  ,IP        ,ACTIM    ,ELTAGE   ,DT       ,
     D     TEMPC   ,RSPTM     ,TEMPR    ,DYNAM    ,ERROR    ,
     E     FEA     ,INIACT    ,NWINC    ,REPAS    ,STEPRD   ,
     F     VERIF   ,
     G     B       ,D         ,DC       ,DISPT    ,DISPC    ,
     H     ELPR    ,EPST      ,EPSC     ,STVBT    ,STVBC    ,
     I     STRST   ,STRSC     ,XYZ      ,VELCTY   )
      DIMENSION
     A     B(NDSE,NDF)   ,D(NDSE,NDSE)   ,DC(NAXES,NAXES),
     B     DISPT(NDF)    ,DISPC(NDF)     ,ELPR(NPRZ)      ,
     C     EPST(NDSE)    ,EPSC(NDSE)     ,STVBT(NSTAT)    ,
     D     STVBC(NSTAT)  ,STRST(NDSE)    ,STRSC(NDSE)     ,
     E     XYZ(NAXES,LNODZ),VELCTY(NDF)
```

Name	Argument Description	Type	Modify
LNODZ	Number of nodes	IV	-
LPTUSR	User supplied constitutive model reference number	IV	-

Name	Argument Description	Type	Modify
NACTVE	Element activation status 0-Standard element 1-activating 2-deactivated	IV	-
NAXES	Number of system axes (dimensions)	IV	-
NDF	Number of degrees of freedom	IV	-
NDSE	Number of continuum stress components at a Gauss point for the stress model type MDL.	IV	-
NEL	Current element number	IV	-
NPRZ	Number of material parameters for the constitutive model	IV	-
NSTAT	Number of state dependent model variables	IV	-
NT6	LUSAS results output channel	IV	-
LSTATM	Length of state variable array	IV	-
IP	Pointer to start of plastic strains	IV	-
ACTIM	Element activation time	RV	-
ELTAGE	Element age	RV	-
DT	Current time step increment (dynamic analysis)	RV	-
TEMPC	Current temperature	RV	-
RSPTM	Current total response time (dynamic analysis)	RV	-
TEMPR	Reference temperature	RV	-
DYNAM	HHT step by step dynamics	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
FEA	Logical flag denoting FEA or external code use	LV	-
INIACT	Logical flag for initial activation of element	LV	-
NWINC	Logical flag, TRUE indicating the start of a new increment	LV	-
REPAS	Logical flag, TRUE for re-pass	LV	-
STEPRD	Step reduction flag	LV	yes
VERIF	Logical incoming argument verification flag	LV	-
B	Strain displacement matrix	RA	-
D	Modulus matrix (D-matrix)	RA	yes
DC	Direction cosine matrix for the local axes system	RA	-
DISPT	Array of displacements at the start of the current increment	RA	-
DISPC	Array of current displacements	RA	-
ELPR	Material properties for current element	RA	-

Name	Argument Description	Type	Modify
EPST	Total strain components at the start of the current increment	RA	-
EPSC	Current total strain components	RA	-
STVBT	State variables at the start of current increment	RA	-
STVBC	Current state variables	RA	yes
STRST	Stress components at the start of the current increment	RA	-
STRSC	Stress components	RA	yes
XYZ	Coordinates for the nodes	RA	-
VELCTY	Current velocity vector	RA	-

SUBROUTINE USRJN1

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutine USRKJN. This subroutine may be utilised by the user but no alterations are permitted.

```
      SUBROUTINE USRJN1
     A(    LNODZ    ,LPTUSR   ,NACTVE   ,NAXES    ,NDF      ,
     B     NDSE     ,NEL      ,NPRZ     ,NPRZC    ,NSTAT    ,
     C     NT6      ,ACTIM    ,ELTAGE   ,DT       ,RSPTM    ,
     D     ERROR    ,FEA      ,INIACT   ,NWINC    ,REPAS    ,
     E     VERIF    ,ARGIN    ,
     F     B        ,D        ,DC       ,DISPT    ,DISPC    ,
     G     ELPR     ,EPST     ,EPSC     ,STVBT    ,STVBC    ,
     H     STRST    ,STRSC    ,XYZ      ,VELCTY   ,TEMPR    ,
     I     TEMPC    )
      DIMENSION
     A     B(NDSE,NDF)   ,D(NDSE,NDSE)  ,DC(NAXES,NAXES),
     B     DISPT(NDF)    ,DISPC(NDF)    ,ELPR(NPRZ)      ,
     C     EPST(NDSE)    ,EPSC(NDSE)    ,STVBT(NSTAT)    ,
     D     STVBC(NSTAT)  ,STRST(NDSE)   ,STRSC(NDSE)     ,
     E     XYZ(NAXES,LNODZ),VELCTY(NDF),TEMPR(NDSE      ,
     F     TEMPC(NDSE)
```

The argument list is the same as that for subroutine USRKJN, except for one argument. The additional argument is the logical variable ARGIN (see continuation line E in the above subroutine statement) that is used in this subroutine. If ARGIN=.TRUE. verification of incoming arguments will be carried out and if ARGIN=.FALSE. verification of outgoing arguments will take place.

SUBROUTINE USRJN2

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutine USRSJN. This subroutine may be utilised by the user but no alterations are permitted.

```
      SUBROUTINE USRJN2
     A(    LNODZ   ,LPTUSR   ,NACTVE   ,NAXES   ,NDF     ,
     B     NDSE    ,NEL      ,NPRZ     ,NSTAT   ,NT6     ,
     C     LSTATM  ,IP       ,ACTIM    ,ELTAGE  ,DT      ,
     D     TEMPC   ,RSPTM    ,TEMPR    ,DYNAM   ,ERROR   ,
     E     FEA     ,INIACT   ,NWINC    ,REPAS   ,STEPRD  ,
     F     VERIF   ,ARGIN    ,
     G     B       ,D        ,DC       ,DISPT   ,DISPC   ,
     H     ELPR    ,EPST     ,EPSC     ,STVBT   ,STVBC   ,
     I     STRST   ,STRSC    ,XYZ      ,VELCTY  )
      DIMENSION
     A     B(NDSE,NDF)   ,D(NDSE,NDSE)   ,DC(NAXES,NAXES),
     B     DISPT(NDF)    ,DISPC(NDF)     ,ELPR(NPRZ)      ,
     C     EPST(NDSE)    ,EPSC(NDSE)     ,STVBT(NSTAT)    ,
     D     STVBC(NSTAT)  ,STRST(NDSE)    ,STRSC(NDSE)     ,
     E     XYZ(NAXES,LNODZ),VELCTY(NDF)
```

The argument list is the same as that for subroutine USRSJN, except for one argument. The additional argument is the logical variable ARGIN (see continuation line F in the above subroutine statement) that is used in this subroutine. If ARGIN=.TRUE. verification of incoming arguments will be carried out and if ARGIN=.FALSE. verification of outgoing arguments will take place.

User Defined Creep Models

Introduction

Several creep models are available in LUSAS, but if the provided models are inappropriate this facility will enable the definition of a creep material using a user-defined creep law.

The user-supplied creep routine USRCRP allows creep laws to be specified that are a function of stress, strain and temperature history.

Creep Laws

The user-supplied subroutine permits creep laws defined as:

$$\dot{\varepsilon}_c = f(q, t, T)$$

where:

$\dot{\varepsilon}_c$	Rate of uniaxial equivalent creep strain	t	Time
q	Equivalent deviatoric stress	T	Temperature

Creep Properties Input

The user creep properties are input in a manner similar to that used for the other creep laws available in LUSAS.

The input for user-defined creep properties consists of a total of **nprzc** creep parameters, the first 3 of which are reserved specifically for use by LUSAS. The temperature is specified should the user wish to utilise temperature dependent properties. Creep properties 4 to **nprzc** must be given in the order expected by the user-supplied routines.

The 3 reserved locations are:

1. Reference temperature (T)
2. Not used at present.
3. Not used at present.

Together with the user creep properties you also need to specify the following integer numbers for the specific creep law defined:

- **ictp** a number which identifies the particular user creep model.
- **nprzc** the total number of creep properties used.

- **nstat** the number of nonlinear creep state variables that are used in the creep model (these variables will be output together with the Gauss point stresses/strains). The value of **nstat** must not be less than 1.

Evaluation of Creep Strains

The following points should be considered when coding the user-supplied routine USRCRP for evaluating the creep strains:

The user-supplied routine must return the increment in creep strain. Further, if implicit integration is to be used, the variation of the creep strain increment with respect to the equivalent stress and with respect to the creep strain increment must also be defined.

If the function involves time dependent state variables, they must be integrated in the user-supplied routine.

If both plasticity and creep are defined for a material, the creep strains will be processed during the plastic strain update. Stresses in the user routine may therefore exceed the yield stress.

Verification

Routine USRVFC is supplied to enable the user to verify the incoming and outgoing arguments to the user-programmed subroutine USRCRP. This subroutine may be utilised by the user but no alterations are permitted.

SUBROUTINE USRCRP

Purpose

Computes the creep strain increment.

The variations of creep strain DCRPDC, DCRPDQ, DCRPDP, DCRPDS are only required for implicit integration. This subroutine may be programmed by the user but the argument list may not be altered in any way.

```
 SUBROUTINE USRCRP
A(   LCTUSR  ,NEL     ,NG      ,NCRPP   ,NT6     ,
B    NSTAT   ,NDSE    ,DISEN   ,DT      ,RSPTIM  ,
C    TEMP    ,PRESS   ,STREQV  ,CRPSTN  ,SWLSTN  ,
D    DCRP    ,DCRPDC  ,DCRPDQ  ,DCRPDS  ,DCRPDP  ,
E    VERIFY  ,ERROR   ,FEA     ,EXPCRP  ,STEPRD  ,
F    TIMEDP  ,
G    STATVC  ,STATVP  ,CRPMDL  ,STRESS  )
 DIMENSION
```

```
A     STATVC(NSTAT) ,STATVP(NSTAT) ,CRPMDL(NCRPP) ,
B     STRESS(NDSE)
```

Name	Argument Description	Type	Modify
LCTUSR	User supplied creep model reference number	IV	-
NEL	Element number	IV	-
NG	Gauss point number	IV	-
NCRPP	Number of parameters defining user creep model	IV	
NT6	LUSAS results output channel	IV	-
NSTAT	Number of creep state dependent model variables	IV	-
NDSE	Number of stress components	IV	-
DISEN	Energy dissipated by inelastic process (not currently used)	RV	-
DT	Time step	RV	-
RSPTIM	Response time t - note time at beginning of step=t, time at end of step=t+dt	RV	-
TEMP	Gauss point temperature	RV	-
PRESS	Current pressure (not currently used)	RV	-
STREQV	Equivalent uniaxial stress (Hill or von Mises)	RV	-
CRPSTN	Creep strain at time t	RV	-
SWLSTN	Swelling strain at time t (not currently used)	RV	-
DCRP	Rate of change of creep strain with time	RV	yes
DCRPDC	Rate of change of creep strain rate with creep increment	RV	yes
DCRPDQ	Rate of change of creep strain rate with equivalent stress	RV	yes
DCRPDS	Rate of change of creep strain rate with swelling strain (not currently used)	RV	-
DCRPDP	Rate of change of creep strain rate with pressure (not currently used)	RV	-
VERIFY	Logical incoming/outgoing argument verification flag	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
FEA	Logical flag for FEA use only	LV	-
EXPCRP	Explicit creep. If .TRUE. use time=RSPTIM for evaluations and if .FALSE. use time=RSPTIM+DT. For an implicit analysis EXPCRP will be .TRUE. in the pre-solution and .FALSE. in the post-solution	LV	-
STEPRD	Step reduction flag	LV	yes

Name	Argument Description	Type	Modify
TIMEDP	Time dependent creep flag	LV	yes
STATVC	On entry: State variables at end of last step On exit: Current state variables	RA	yes
STATVP	State variables at end of last step	RA	-
CRPMDL	Creep model parameters	RA	-
STRESS	Current stress tensor	RA	-

SUBROUTINE USRVFC

Purpose

Verifies the incoming and outgoing arguments to the user-programmed subroutine USRCRP. This subroutine may be utilised by the user but no alterations are permitted.

```
      SUBROUTINE USRVFC
     A(    LCTUSR  ,NEL     ,NG      ,NCRPP   ,NT6     ,
     B     NSTAT   ,NDSE    ,DISEN   ,DT      ,RSPTIM  ,
     C     TEMP    ,PRESS   ,STREQV  ,CRPSTN  ,SWLSTN  ,
     D     DCRP    ,DCRPDC  ,DCRPDQ  ,DCRPDS  ,DCRPDP  ,
     E     VERIFY  ,ERROR   ,FEA     ,EXPCRP  ,STEPRD  ,
     F     TIMEDP  ,MODE    ,
     G     STATVC  ,STATVP  ,CRPMDL  ,STRESS     )
      DIMENSION
     A     STATVC(NSTAT) ,STATVP(NSTAT) ,CRPMDL(NCRPP) ,
     B     STRESS(NDSE)
```

The argument list is the same as that for subroutine USRCRP, except for the additional integer variable MODE (see continuation line F in the above subroutine statement). This variable is used in this subroutine: MODE=1 is for verification of incoming arguments, MODE=(any other value) is for verification of outgoing arguments.

User Defined Damage Models

Introduction

The user-supplied damage routine USRDAM permits external computation of the damage variable and its derivative with respect to the current elastic complementary energy norm.

The temperature, the damage threshold at the previous converged position, the damage model parameters and the damage state variables at the current and previous converged positions are available for this computation.

This facility allows the user to specify a damage law for a particular material if the damage models available in LUSAS are inappropriate.

Damage Variable

The damage variable is used to define the degradation of the elastic modulus matrix. This means that the effective stress vector may be expressed as:

$$\left\{\sigma_e^t\right\} = \frac{1}{1-d(t)}\left\{\sigma_c^t\right\}$$

$\left\{\sigma_e^t\right\}$ Effective stress vector at time t.

$\left\{\sigma_c^t\right\}$ Cauchy stress vector at time t.

$d(t)$ Damage variable at time t. Note that for no damage, $d(t) = 0$.

Further information on the use of damage models can be found in the *LUSAS Theory Manual*.

Damage Properties Input

The user damage properties are input in a manner similar to that used for the other damage laws available in LUSAS.

The user damage properties input consists of a total of **nprzd** damage parameters, the first of which is reserved specifically for LUSAS use. This is the temperature that is specified, should the user wish to utilise temperature dependent properties. Damage properties 2 to **nprzd** must be supplied by the user in the order required by the user-supplied routines.

The reserved location is:

Reference temperature. (T)

Together with the user damage properties, the following integer numbers for the specific damage law defined also need to be specified.

- **idtp** Number which identifies the particular user damage model.
- **nprzd** Total number of damage properties used.
- **nstat** Number of nonlinear state variables that are used in the damage model (these variables will be output together with the Gauss point stresses/strains). The value of **nstat** must not be less than 1.

Verification

Routine USRVFD is supplied to enable the user to verify the incoming and outgoing arguments to the user-programmed subroutines USRDAM. This subroutine may be utilised by the user but no alterations are permitted.

SUBROUTINE USRDAM

Purpose

Computes the damage variable and its derivative with respect to the current elastic complementary energy norm. This subroutine may be programmed by the user but the argument list may not be altered in any way.

```
 SUBROUTINE USRDAM
A(    LDTUSR   ,NEL      ,NG       ,NDAMP    ,NT6      ,
B     NSTAT    ,TEMP     ,TAUC     ,HDAMA    ,DDAMA    ,
C     TDAMA    ,ERROR    ,VERIFY   ,FEA      ,
D     DAMCUR   ,DAMPRV   ,DAMMDL   )
 DIMENSION
A     DAMCUR(NSTAT)  ,DAMPRV(NSTAT)  ,DAMMDL(NDAMP)
```

Name	Argument Description	Type	Modify
LCTUSR	User supplied damage model reference number	IV	-
NEL	Element number	IV	-
NG	Gauss point number	IV	-
NDAMP	Number of parameters defining user damage model	IV	-
NT6	LUSAS results output channel	IV	-

Name	Argument Description	Type	Modify
NSTAT	Number of damage state dependent model variables		
TEMP	Gauss point temperature	RV	-
TAUC	Norm of current elastic complementary energy	RV	-
HDAMA	Derivative of damage variable with respect to TAUC	RV	yes
DDAMA	Damage variable	RV	yes
TDAMA	Damage threshold at previous converged position	RV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
VERIFY	Logical incoming/outgoing argument verification flag	LV	-
FEA	Logical flag for FEA use only	LV	-
DAMCUR	Current damage state variables	RA	yes
DAMPRV	Damage state variables at end of last step	RA	-
DAMMDL	Damage model parameters	RA	-

SUBROUTINE USRVFD

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutine USRDAM. This subroutine may be utilised by the user but no alterations are permitted.

```
      SUBROUTINE USRVFD
     A(    LDTUSR  ,NEL     ,NG      ,NDAMP   ,NT6     ,
     B     NSTAT   ,TEMP    ,TAUC    ,HDAMA   ,DDAMA   ,
     C     TDAMA   ,ERROR   ,VERIFY  ,FEA     ,MODE    ,
     D     DAMCUR  ,DAMPRV  ,DAMMDL  )
      DIMENSION
     A     DAMCUR(NSTAT) ,DAMPRV(NSTAT) ,DAMMDL(NDAMP)
```

The argument list is the same as that for subroutine USRDAM, except for the additional integer variable MODE (see continuation line C in the above subroutine statement). This variable is used in this subroutine: MODE=1 is for verification of incoming arguments, MODE=(any other value) is for verification of outgoing arguments.

User Defined Friction Models

Introduction

The user-supplied routine USRSLF permits a nonlinear friction law to be utilised in a slideline analysis. The friction law may be a function of the surface temperature, the relative velocities and/or accelerations of the adjacent surfaces and a set of user defined friction parameters. The friction law can be used with the normal interface force to return an allowable tangential frictional force.

Temperature dependent parameters are entered using the standard tabular input. All the values in the slideline properties table are interpolated at the temperature of the point of contact on the contacted surface prior to passing into the user routine.

Nonlinear Friction Law

The nonlinear friction law can take the form:

$$F_{allow} = f(N, \mu_1, \mu_2, \ldots \mu_i, v, a, T)$$

where:

F_{allow}	Allowable tangential frictional force	v	Relative velocities of the contacting surfaces
N	Normal interface force	α	Relative accelerations of the contacting surfaces
μ_i	User defined friction parameters	T	Temperatures of contacting surfaces

The components of velocity and acceleration supplied to the user routine are in the direction of the tangential frictional force on the slideline surface being processed.

Note that in the routine USRSLF, the adjacent surface is the current surface being processed. To provide further useful information, the current response time, the contacting node number and the node numbers defining the current segment of the adjacent surface are also passed into USRSLF.

Nonlinear Friction Parameter Input

The nonlinear friction parameters are input using the SLIDELINE PROPERTIES USER data section.

Input for the SLIDELINE PROPERTIES USER data section follows the same form as the standard SLIDELINE PROPERTIES data section, except the number of friction

parameters (nfric) must be defined. The friction parameters must be supplied in the order required by the user-supplied routines.

Verification

Routine USRVSF is supplied to enable the user to verify the incoming and outgoing arguments to the user-programmed subroutines USRSLF. This subroutine may be utilised but no alterations are permitted.

SUBROUTINE USRSLF

Purpose

Computes the allowable tangential force using a nonlinear friction law. This subroutine may be programmed by the user but the argument list may not be altered in any way.

```
 SUBROUTINE USRSLF
A(    ISPROP   ,NODC     ,NFRIC    ,NT6      ,ISURF    ,
B     JSURF    ,NDSEG    ,NPROPS   ,TMPND    ,TMPSF    ,
C     VNOD     ,VSUR     ,ANOD     ,ASUR     ,FORCEN   ,
D     FTALLW   ,RSPTIM   ,VERIFY   ,ERROR    ,FEA      ,
E     NODSEG   ,SLPROP   )
 DIMENSION
A     NODSEG(NSEGN)           ,SLPROP(NPROPS)
```

Name	Argument Description	Type	Modify
ISPROP	Slideline property assignment number	IV	-
NODC	Contact node number	IV	-
NFRIC	Number of friction parameters	IV	-
NT6	LUSAS output channel	IV	-
ISURF	Surface number containing contact node	IV	-
JSURF	Cntacted surface number	IV	-
NDSEG	Number of nodes on the current segment of the contacted surface	IV	-
TMPND	Current temperature at contacting node	RV	-
TMPSF	Current temperature at contact point on adjacent surface	RV	-
VNOD	Velocity component of contacting node in direction of total tangential force on adjacent surface	RV	-

Name	Argument Description	Type	Modify
VSUR	Velocity component of adjacent surface in direction of total tangential force	RV	-
ANOD	Acceleration component of contacting node in direction of total tangential force on adjacent surface	RV	-
ASUR	Acceleration component of adjacent surface in direction of total tangential force	RV	-
FORCEN	Magnitude of the normal force	RV	-
FTALLW	Allowable tangential force	RV	yes
RSPTIM	Current response time	RV	-
VERIFY	Logical flag for verification of arguments	LV	-
ERROR	Logical flag for fatal error and program termination	LV	yes
FEA	For LUSAS/FEA internal use only	LV	-
NODSEG	Node numbers for segment of adjacent surface	IA	-
SLPROP	Slideline properties evaluated at contacted surface temperature TMPSF	RA	-

SUBROUTINE USRVSF

Purpose

Verify the incoming and outgoing arguments to the user-programmed subroutine USRSLF. This subroutine may be utilised by the user but no alterations are permitted.

```
 SUBROUTINE USRVSF
A(   ISPROP  ,NODC    ,NFRIC   ,NT6     ,ISURF   ,
B    JSURF   ,NDSEG   ,NPROPS  ,TMPND   ,TMPSF   ,
C    VNOD    ,VSUR    ,ANOD    ,ASUR    ,FORCEN  ,
D    FTALLW  ,RSPTIM  ,VERIFY  ,ERROR   ,FEA     ,
E    MODE    ,
F    NODSEG  ,SLPROP  )
 DIMENSION
A    NODSEG(NSEGN)          ,SLPROP(NPROPS)
```

The argument list is the same as that for subroutine USRSLF, except for the additional integer variable MODE (see continuation line E in the above subroutine statement). This variable is used in this subroutine: MODE=1 is for verification of incoming arguments, MODE=(any other value) is for verification of outgoing arguments.

User-Defined Rate of Internal Heat Generation

Introduction

The user-supplied subroutine USRRHG permits the user to define the way in which internal heat is generated in a thermal analysis. This can be defined to be a function of temperature, time and chemical reaction. The parameters used to control the chemical reaction are specified under the RIHG USER data chapter. In a thermo-mechanical coupled analysis, variables defining the current degree and rate of chemical reaction (or cure) may be transferred to the structural analysis where they can be accessed in the user interface routines USRKDM and USRSTR. The modulus matrix and stress computations may then become a function of degree or rate of cure.

User RIHG Parameter Input

The parameters controlling the rate of internal heat generation are specified under the user RIHG chapter in a similar manner to the way standard RIHG parameters are defined in LUSAS. The RIHG User data section takes the following form:

```
TEMPERATURE LOAD CASE
RIHG USER  n
TABLE  ilod
<P_i>_i=1,n
TEMPERATURE LOAD ASSIGNMENTS
RIHG USER
L   N   ilod
```

Where **n** is the number of parameters to be defined, **ilod** the table identifier, $\mathbf{P_i}$ the RIHG control parameters, one of which could be a reference temperature, **L** the element number and **N** the node number of that element for which the parameters are to be applied. Note that the table could consist of many lines of parameters each one relating to a particular reference temperature. The complete table is accessible in the user routine USRRHG so that temperature dependent control parameters can be specified if necessary.

Evaluation of RIHG

The following points should be considered when coding the user-supplied routine USRRHG for evaluating the rate of internal heat generation:

1. The user supplied routine USRRHG is called from within the element gauss point loop.
2. The routine must return the rate of internal heat generation at the nodes stored in the vector RIHG. In addition, the degree and rate of cure at the nodes can also be computed, stored in array DOCCND and returned. As this routine is called from within a gauss point loop and the information required to compute the nodal values

is always available for an element, the nodal values need only be computed and returned on the first pass for each element, e.g. when the gauss point loop counter, NG=1.

3. The current degree and rate of cure (or some other chemical reaction) can also be computed at the gauss points on each pass through the routine and stored in vector DOCCGP. The values at the start of the time step are also available.
4. In a thermo-mechanical coupled analysis, the gauss point cure values (or any other chemical reaction which influences mechanical behaviour) may be transferred to the structural analysis and are accessible in the user defined material routines USRKDM and USRSTR.

Verification

Routine USRVFR is supplied to enable the user to verify the incoming and outgoing arguments to the user-programmed subroutines USRRHG. This subroutine may be utilised by the user but no alterations are permitted.

SUBROUTINE USRRHG

Purpose

Computes the rate of internal heat generation at nodes, RIHG, the degree of cure at nodes, DOCCND and at gauss points, DOCCGP. This subroutine may be programmed by the user but the argument list may not be altered in any way.

```
 SUBROUTINE USRRHG
A(   NEL     ,LNODZ   ,NTAB    ,NRHGD   ,NG      ,
B    INC     ,NT      ,NAXES   ,NGP     ,NT6     ,
C    NDOC    ,DT      ,RSPTM   ,DINT    ,TEMPIN  ,
D    ACTIM   ,ELTAGE  ,USRGPC  ,USRGPP  ,ERROR   ,
E    BOUNDS  ,PRESOL  ,VERIF   ,NWINC   ,FEA     ,
F    TMPFLG  ,REPAS   ,
G    TINIT   ,TEMPC   ,TEMPP   ,TEMPI   ,NTABLD  ,
H    SHAPES  ,XYZ     ,DOCCGP  ,DOCPGP  ,DOCIGP  ,
I    RIHG    ,RHGDAT  ,XYZN    ,DOCCND  ,DOCPND  ,
J    DOCIND  ,USRNDC  ,USRNDP  )
 DIMENSION
A    TINIT(LNODZ)          ,TEMPC(LNODZ)          ,
B    TEMPP(LNODZ)          ,TEMPI(LNODZ)          ,
C    NTABLD(LNODZ)         ,SHAPES(LNODZ)         ,
D    XYZ(NAXES)            ,DOCCGP(NDOC)          ,
```

```
E     DOCPGP(NDOC)            ,DOCIGP(NDOC)              ,
F     RIHG(LNODZ)          ,RHGDAT(NRHGD,NTAB,LNODZ),
G     XYZN(NAXES,LNODZ)       ,DOCCND(NDOC,LNODZ)     ,
H     DOCPND(NDOC,LNODZ)      ,DOCIND(NDOC,LNODZ)     ,
I     USRNDC(LNODZ)           ,USRNDP(LNODZ)
```

Name	Argument Description	Type	Modify
NEL	Current element number	IV	-
LNODZ	Number of element nodes	IV	-
NTAB	Maximum number of lines in any RIHG USER table	IV	-
NRHGD	Maximum number of parameters in any line of RIHG USER data	IV	-
NG	Current Gauss point number	IV	-
INC	Current increment number	IV	-
NT	Current time step number	IV	-
NAXES	Number of system axes (dimensions)	IV	-
NGP	Number of gauss points	IV	-
NT6	LUSAS results output channel	IV	-
NDOC	First dimension of degree of cure arrays (=2)	IV	-
DT	Current time step increment	RV	-
RSPTM	Current total response time	RV	-
DINT	Characteristic length or area of the current Gauss point	RV	-
TEMPIN	Temperature at the current gauss point - interpolated from nodal values (can be used if TMPFLG.is true - currently only applicable for composite brick elements)	RV	-
ACTIM	Time of element activation	RV	
ELTAGE	Element age on loading	RV	
USRGPC	User defined gauss point value (current)	RV	yes
USRGPP	User defined gauss point value (start of time step)	RV	
ERROR	Logical flag for fatal error and program termination	LV	yes
BOUNDS	Logical to error if the current temperature is outside the bounds of a table of values (OPTION 227)	LV	-
PRESOL	Logical flag, TRUE for presolution (forming thermal stiffness) and FALSE in postsolution (computation of thermal gradients and flows)	LV	-

Name	Argument Description	Type	Modify
VERIF	Logical incoming/outgoing argument verification flag	LV	-
NWINC	Logical flag, TRUE denoting the start of a new increment	LV	-
FEA	Logical flag denoting FEA or external code use (FOR FEA INTERNAL USE ONLY)	LV	-
TMPFLG	Logical flag, if true indicates that the current gauss point temperature has been interpolated from nodal values and is available in the arguments list	LV	-
REPAS	Logical flag, TRUE denoting a re-pass	LV	-
TINIT	Initial nodal temperatures - only applicable on very first pass when specified as TMPE or TEMP loading case	RA	-
TEMPC	Current nodal temperatures	RA	-
TEMPP	Nodal temperatures at previous increment or time step	RA	-
TEMPI	Nodal temperatures at previous iteration	RA	-
NTABLD	Number of lines in the load table for each node	IA	-
SHAPES	Shape function values for current gauss point	RA	-
XYZ	Current gauss point coordinates	RA	-
DOCCGP	Current degree of cure at current gauss point	RA	yes
DOCPGP	Degree of cure at previous increment or time step at current gauss point	RA	-
DOCIGP	Degree of cure at previous iteration at current gauss point	RA	-
RIHG	Rate of internal heat generation at nodes	RA	yes
RHGDAT	Input parameters defined in RIHG USER data chapter	RA	-
XYZN	Nodal coordinates	RA	-
DOCCND	Current degree of cure at nodes	RA	yes
DOCPND	Nodal degree of cure at previous increment or time step	RA	-
DOCIND	Nodal degree of cure at previous iteration	RA	-
USRNDC	User defined nodal values (current)	RA	yes
USRNDP	User defined nodal values (start of time step)	RA	

SUBROUTINE USRVFR

Purpose

Verifies the incoming and outgoing arguments to the user-programmed subroutine USRRHG. This subroutine may be utilised by the user but no alterations are permitted.

```
      SUBROUTINE USRVFR
     A(    NEL     ,LNODZ   ,NTAB    ,NRHGD   ,NG      ,
     B     INC     ,NT      ,NAXES   ,NGP     ,NT6     ,
     C     NDOC    ,DT      ,RSPTM   ,DINT    ,TEMPIN  ,
     D     ACTIM   ,ELTAGE  ,USRGPC  ,USRGPP  ,ERROR   ,
     E     BOUNDS  ,PRESOL  ,VERIF   ,NWINC   ,FEA     ,
     F     TMPFLG  ,REPAS   ,ARGIN   ,
     G     TINIT   ,TEMPC   ,TEMPP   ,TEMPI   ,NTABLD  ,
     H     SHAPES  ,XYZ     ,DOCCGP  ,DOCPGP  ,DOCIGP  ,
     I     RIHG    ,RHGDAT  ,XYZN    ,DOCCND  ,DOCPND  ,
     J     DOCIND  ,USRNDC  ,USRNDP  )
      DIMENSION
     A     TINIT(LNODZ)           ,TEMPC(LNODZ)          ,
     B     TEMPP(LNODZ)           ,TEMPI(LNODZ)          ,
     C     NTABLD(LNODZ)          ,SHAPES(LNODZ)         ,
     D     XYZ(NAXES)             ,DOCCGP(NDOC)          ,
     E     DOCPGP(NDOC)           ,DOCIGP(NDOC)          ,
     F     RIHG(LNODZ)          ,RHGDAT(NRHGD,NTAB,LNODZ),
     G     XYZN(NAXES,LNODZ)      ,DOCCND(NDOC,LNODZ)    ,
     H     DOCPND(NDOC,LNODZ)     ,DOCIND(NDOC,LNODZ)    ,
     I     USRNDC(LNODZ)          ,USRNDP(LNODZ)
```

The argument list is almost the same as that of subroutine USRRHG, except for the additional logical ARGIN (see continuation line F in the above subroutine statement). This variable is used in this subroutine. If ARGIN=.TRUE. verification of incoming arguments will be carried out and if ARGIN=.FALSE. verification of outgoing arguments will take place.

Utility Routines

Introduction

In addition to the user-programmable subroutines, access is available to the following LUSAS utility subroutines and functions. A full description of the operation and arguments of each of these routines can be found on the pages that follow.

Name	Routine Function and Description
DVSTSN	Form deviatoric stress matrix from total stress matrix and evaluate the first stress invariant
INTPRP	Interpolate the current material properties from the table
KMVEQS	Transform stress/strain vector to equivalent matrix or vice versa
PRNSTR	Compute the principal values of stress/strain vector in 2-D
PRSTR3	Compute the principal values of stress/strain vector in 3-D
TEGV2	Compute eigenvalues and eigenvectors of a 2*2 matrix
MTEGV3	Compute eigenvalues and eigenvectors of a 3*3 matrix
MTINV2	Invert a 2*2 matrix explicitly
MTINV3	Invert a 3*3 matrix using Cramer's rule
MTINVT	Invert a matrix of any order using Gaussian elimination
KTRNLG	Transform stress/strains between local and global systems at a point

In order to avoid the duplication of existing LUSAS subroutine names, any externally developed subroutines (at a lower level to the user-programmable subroutines) must adhere to the naming convention:

```
SUBROUTINE USR___
```

It is also recommended that site dependent constraints regarding subroutine name length, variable names, and machine precision be considered when developing external FORTRAN code.

SUBROUTINE DVSTSN

Purpose

Calculates the deviatoric stress matrix and then evaluates the first stress invariant. This subroutine may be utilised by the user but no alterations are permissible.

```
 SUBROUTINE DVSTSN
A(    NMSMX    ,SMEAN    ,
B     SMX      ,DEVSMX   )
 DIMENSION
```

```
      A     SMX(NMSMX,NMSMX)          ,DEVSMX(NMSMX,NMSMX)
```

Name	Argument Description	Type	Modified
NMSMX	Dimensions of matrices (always=3)	IV	-
SMEAN	Mean stress/strain	RV	yes
SMX	Stress/strain matrix	RA	-
DEVSMX	Deviatoric stress tensor	RA	yes

SUBROUTINE INTPRP

Purpose

Interpolates temperature dependent properties from a table. The temperature property values within which the temperature load lies are found and the column numbers of these properties in the element material property table array are extracted. These properties are then interpolated as a linear variation of temperature and stored in the element material property array. This subroutine may be utilised by the user but no alterations are permissible.

```
      SUBROUTINE INTPRP
     A(    NEL      ,IMAT     ,LPTP     ,NPRZ     ,NTAB     ,
     B     NG       ,ILOC     ,TEMP     ,BOUNDS   ,ERROR    ,
     C     ELPR     ,ELPRT    )
      DIMENSION
     A     ELPR(NPRZ)               ,ELPRT(NPRZ,NTAB)
```

Name	Argument Description	Type	Modified
NEL	Element number	IV	-
IMAT	Material assignment reference number	IV	-
LPTP	Material model reference number	IV	-
NPRZ	Maximum number of properties for defined property type	IV	-
NTAB	Number of lines of property data in table	IV	-
NG	Gauss point number	IV	-
ILOC	Location of temperature property in ELPR or ELPRT for material defined	IV	yes
TEMP	Temperature at this point	RV	-
BOUNDS	Flag for error if temperature out of bounds	LV	-

Name	Argument Description	Type	Modified
ERROR	Logical flag for fatal error and program termination	LV	yes
ELPR	Interpolated element material properties	RA	yes
ELPRT	Table of temperature dependent material properties	RA	-

SUBROUTINE KMVEQS

Purpose

This subroutine sets up the stress/strain tensor as a 3*3 matrix or NDSE*1 vector for any stress model. This subroutine may be utilised by the user but no alterations are permissible.

```
 SUBROUTINE KMVEQS
A(   MDL      ,NDSE     ,NMS      ,STRFLG   ,MTXFLG   ,
B    SM       ,SV       )
 DIMENSION
A    SM(NMS,NMS)    ,SV(NDSE)
```

Name	Argument Description	Type	Modified
MDL	Reference code for stress model type	IV	-
NDSE	Number of continuum stresses/strains at a point	IV	-
NMS	Dimension of stress/strain matrix	IV	-
STRFLG	Flag indicating stress or strain to be used, .TRUE.=Stress, .FALSE.=Strain	LV	-
MTXFLG	Flag indicating matrix or vector form is required, .TRUE.=Matrix, .FALSE.=Vector	LV	-
SM	Stress/strain matrix	RA	yes
SV	Stress/strain vector	RA	yes

SUBROUTINE PRNSTR

Purpose

Computes the principal stresses or strains for 2-dimensional problems using Mohr's circle. This subroutine may be utilised by the user but no alterations are permissible.

```
 SUBROUTINE PRNSTR
```

```
      A(    S       ,SMAX     ,SMIN     ,ANGD     ,MODE     )
       DIMENSION
      A     S(3)
```

Name	Argument Description	Type	Modified
S	Vector in the form Sx, Sy, Sxy	RA	-
SMAX	Maximum principal value	RV	yes
SMIN	Minimum principal value	RV	yes
ANGD	Angle of maximum principal value in degrees clockwise from the X axes	RV	yes
MODE	Flag indicating stress or strain, 1 = Stress, 2 = Strain	IV	-

SUBROUTINE PRSTR3

Purpose

This subroutine finds the principal stresses/strains in 3D by computing the eigenvectors of a symmetric tri-diagonal matrix using inverse iteration. This subroutine may be utilised by the user but no alterations are permissible.

```
       SUBROUTINE PRSTR3
      A(    NDSE     ,NPRIN    ,MDL      ,STRFLG
      B     STRESS   ,STRSP    ,SVEC     )
       DIMENSION
      A     STRESS(NDSE)             ,STRSP(NPRIN)            ,
      B     SVEC(NPRIN,NPRIN)
```

Name	Argument Description	Type	Modified
NDSE	Number of continuum stress/strain components	IV	-
NPRIN	Dimension of tri-diagonal matrix (must equal 3)	IV	-
STRESS	Stress array: Sx, Sy, Sz, Sxy, Syz, Sxz	RA	-
STRSP	Principal stress array	RA	yes
SVEC	Final eigenvectors	RA	yes

SUBROUTINE MTEGV2

Purpose

Computes the explicit solution in terms of eigenvalues and eigenvectors of the 2-dimensional eigen problem. This subroutine may be utilised by the user but no alterations are permissible.

```
      SUBROUTINE MTEGV2
     A(    IS       ,CM       ,EVALUS   ,EVECTS   ,ERROR    )
      DIMENSION
     A     CM(IS,IS)      ,EVALUS(IS)     ,EVECTS(IS,IS)
```

Name	Argument Description	Type	Modified
IS	Dimension of eigenproblem (must = 2)	IV	-
CM	Coefficient matrix (must be positive definite)	RA	-
EVALUS	Array of eigenvalues	RA	yes
EVECTS	Matrix of eigenvectors	RA	yes
ERROR	Flag returned .TRUE. on error	LV	yes

SUBROUTINE MTEGV3

Purpose

Compute the solution in terms of eigenvalues and eigenvectors of a symmetric 3-dimensional eigen problem using Jacobi rotation. This subroutine may be utilised by the user but no alterations are permissible.

```
      SUBROUTINE MTEGV3
     A(    IS       ,CM       ,EVALUS   ,EVECTS   )
      DIMENSION
     A     CM(IS,IS)      ,EVALUS(IS)     ,EVECTS(IS,IS)
```

Name	Argument Description	Type	Modified
IS	Dimension of eigenproblem (must = 3)	IV	-
CM	Coefficient matrix (must be positive definite)	RA	-
EVALUS	Array of eigenvalues	RA	yes
EVECTS	Matrix of eigenvectors	RA	yes

SUBROUTINE MTINV2

Purpose

Invert a 2*2 matrix explicitly. A mode switch is available so that only the determinant can be computed. This subroutine may be utilised by the user but no alterations are permissible.

```
 SUBROUTINE MTINV2
A(   A       ,B       ,DET     ,IS      ,MODE    ,
B    IERR    )
 DIMENSION
A    A(IS,IS)      ,B(IS,IS)
```

Name	Argument Description	Type	Modified
A	Matrix to be inverted	RA	-
B	Inverse matrix	RA	yes
DET	Determinant of A	RV	yes
IS	Dimension of matrix (must equal 2)	IV	-
MODE	Mode switch, 1= evaluate determinant only	IV	-
IERR	Error flag, returned non-zero on error	IV	yes

SUBROUTINE MTINV3

Purpose

Invert a 3*3 matrix explicitly. A mode switch is available so that only the determinant can be computed. This subroutine may be utilised by the user but no alterations are permissible.

```
 SUBROUTINE MTINV3
A(   A       ,B       ,DET     ,IS      ,MODE    ,
B    IERR    )
 DIMENSION
A    A(IS,IS)      ,B(IS,IS)
```

Name	Argument Description	Type	Modified
A	Matrix to be inverted	RA	-
B	Inverse matrix	RA	yes

Name	Argument Description	Type	Modified
DET	Determinant of A	RV	yes
IS	Dimension of matrix (must = 3)	IV	-
MODE	Mode switch, 1=evaluate determinant only	IV	-
IERR	Error flag, returned nonzero on error	IV	yes

SUBROUTINE MTINVT

Purpose

Invert a N*N matrix (or a sub-matrix) by using the Gauss-Jordan method. This subroutine may be utilised by the user but no alterations are permissible.

```
      SUBROUTINE MTINVT
     A(       A        ,IS      ,NA      )
      DIMENSION
     A        A(IS,IS)
```

Name	Argument Description	Type	Modified
A	On input : matrix to be inverted	RA	yes. On exit: the inverse of the original matrix
IS	Dimension of matrix	IV	-
NA	The dimension of the sub-matrix that must be inverted	IV	-

SUBROUTINE KTRNLG

Purpose

Transform local stresses or strains at a point to global values. This subroutine may be utilised by the user but no alterations are permissible.

```
      SUBROUTINE KTRNLG
     A(    NEL      ,NDSE     ,NAXES    ,MODE     ,TYPE     ,
     B     MDL      ,
     C     DRCGGP   ,STRORI   ,STRNEW   )
      DIMENSION
```

```
A     DRCGGP(NAXES,NAXES)     ,STRORI(NDSE)          ,
B     STRNEW(NDSE)
```

Name	Argument Description	Type	Modified
NEL	Element number	IV	-
NDSE	Number of stresses/strains at a point	IV	-
NAXES	Number of axes for a problem	IV	-
MODE	If .TRUE. then local to global else global to local	LV	-
TYPE	Logical for results type .TRUE. for stresses otherwise strains	LV	-
MDL	Stress model number	IV	-
DRCGGP	Direction cosines defining local Cartesian axes	RA	-
STRORI	Original stresses or strains	RA	-
STRNEW	Transformed stresses or strains	RA	yes